4050 SERIES APPLICATIONS LIBRARY

MASTER TAPE #19

FILE	ABSTRACT	TITLE	MARKED BYTES	BYTES USED FOR PROGRAM	COMMENTS
1	51/07-0102/0	GANTT CHART	1792	1537	Requires pre-MARKed data files; requires dedicated tape
2	"	"	9472	9297	
3	"	"	11264	11020	
4	"	"	19200	18891	
5	"	"	768	33	
6	"	"	1280	925	
7	"	"	6912	6740	
8	51/00-0103/0	CAPITAL BUDGET	17152	17043	Automatically marks data files; requires dedicated tape
9	"	"	768	3	
10	51/00-5506/0	3D FUNCTION PLOT	5632	5341	
11	51/07-6108/1	RECORDKEEP II (REVISION)	4096	3744	Must be first 14 files on tape or recode
12	"	"	5888	5773	
13	"	"	2816	2620	
14	"	"	6912	6644	
15	"	"	5376	5192	
16	"	"	6656	6372	
17	"	"	5120	5000	
18	"	"	4352	4115	
19	"	"	3328	3172	
20	"	"	7168	6961	
21	"	"	4352	4016	
22	"	"	3584	3272	
23	"	"	3072	2942	
24	"	"	2560	2238	
25	51/00-6117/0	TEKTRONIX/HEWLETT PACKARD	1280	1128	
26	"	488 INTERFACE	1792	1592	
27	51/00-6118/0	20mA CURRENT LOOP INTERFACE	1536	1435	
28	51/00-8012/1	LEROY CHARACTER GENERATOR (REVISION)	5888	5744	
29	"	"	6912	6628	
30	51/00-8036/0	SPIDER'S WEB CHARACTER GENERATOR	7936	7636	
31	51/00-8037/0	AUTOMATIC HARDCOPY FILE LISTINGS	3840	3502	
32	51/00-8038/0	TEXT EDITOR	6400	6204	Requires pre-MARKed file
33	51/00-8040/0	VARIABLE FONT PRESENTATION AID	7680	7508	Must be first 9 files on tape or recode
34	"	"	3072	2880	
35	"	"	7936	7702	
36	"	"	3072	2880	
37	"	"	7680	7432	
38	"	"	3072	2880	
39	"	"	4864	4744	
40	"	"	3072	2880	
41	"	"	5376	5200	
42	51/00-8041/0	BINARY DATA FILE DUPLICATOR	6144	5846	Requires pre-MARKed file
43	51/00-9537/0	OVERLAY DRAWING PROGRAM	5632	5483	Optional-pre-MARKed data files
44	"	"	768	413	
45	"	"	768	427	
46	51/07-9539/0	ENHANCED PIE CHART	9984	9268	Requires data files; must be first 7 files on tpae or recode
47	"	"	1792	1054	
48	"	"	2048	1472	
49	"	"	7424	6721	
50	"	"	17152	16552	
51	"	"	1792	1058	
52	"	"	1536	855	

File 23

#1

TEKTRONIX®

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0
ORIGINAL DATE 11/26/76 / REVISION DATE	MEMORY REQUIREMENT Minimum
AUTHOR	PERIPHERALS
Dan Taylor, G. S. Application Development	A Printer os Optional

ABSTRACT

This program reads a BASIC program from tape and produces an alphabetized table of the variables used in the program, the same as 51/00-8003/0 does. It also produces a cross-reference for each variable used which shows the BASIC line numbers where that variable is used and indicates if a value is assigned to that variable in that line of code. The BASIC program may be stored on multiple tape files.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

Program Limitations

The tape files (s) to be analyzed must be valid 4051 BASIC programs. The programs should have been stored on tape with the SAVE command.

An asterisk is normally printed with the line number when a value is assigned to the variable in that line of code. (eg X=2.7 INPUT X, READ X, etc.) But an asterisk is not printed if a value is assigned to an element of an array (eg X(2) = 4.6).

Your program may be too large for all the variables to be listed at once. If this happens the following message will be printed:

STORAGE FILLED IN THE MIDDLE OF FILE # 1. THE PROGRAM WILL CONTINUE WITH FILE # 1 AFTER THE FOLLOWING INFORMATION IS PRINTED:

VARIABLES:

A

A$

B$

C$

' etc. '

' '

' '

The larger size memory you have the slower storage will be filled (see Methodology).

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

Methodology

The input files are read line by line and searched for variable names. When a variable is found its name, line number, and file number are saved at the end of the string N$. An asterisk is also put in N$ if a value is assigned to the variable in this line. A total of 15 characters are required in N$ for each occurrence of every variable.

If N$ fills up before all your files are analyzed then the information in it is immediately printed. The cross-referencing then continues. In this case your cross-reference would be in two parts.

How soon N$ fills depends on the machine size you have:

16K - 390 variable occurrences

24K - 930 variable occurrences

32K - 1480 variable occurrences

The information stored in N$ is organized as a collection of linked lists. There is one list for each variable you use. The array A (A(26,12)) consists of pointers into N$. For example, A (2,2) might be 6181.0625. This means that:

1 BØ is used

2 4096 (=.0625 x 2 ↑ 16) is the start in N$ of the linked list for BØ.

3 6181 is the end in N$ of the linked list for BØ.

A (j,k) = Ø means that that variable is not used at all.

A (1,1) = A
1,2 = AØ
:
:
:
1,1Ø = A9
1,12 = A$

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

Operating Hints

If you wish to use this program with a printer you must change 3 lines of code:

FROM	TO
1030 F4 = 6	1030 F4 = 10
1040 FS = 1.0E + 300	1040 FS = 60
1050 JS = 32	1060 JS = 41

F4 = the number of items to print per line (each require 12 spaces)

FS = the number of lines to print before doing a form feed (control L)

JS = device address of where the cross-reference information will be printed (32 is for the screen)

Line 1020 is a REM which also tells you about these changes.

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

Operating Instructions

After the program is in the machine type RUN:

CROSS-REFERENCE AND LIST PROGRAM VARIABLES

NOTE: YOUR PROGRAM MAY BE ON SEVERAL TAPE FILES.

Begin with File Number = 29

End with File Number = 30

FILE # 29

FILE # 30

When the last file has been processed*(30 in this example) the program prints a table which gives you an overview of which variables were used:

* or when N$ is filled - see Limitations

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

VARIABLES:

AØ					A6			A9	
									B$
DØ			D3						D$
EØ	E1								
	F1	F2							
HØ	H1	H2							H$
	L1							L9	
MØ									
			N3						
	P1								P$
RØ	R1		R3	R4				R9	
SØ		S2	S3	S4	S5	S6	S7	S9	S$
TØ									T$
	U1	U2	U3	U4		U6	U7	U8	U9
	V1	V2	V3	V4	V5	V6	V7	V8	
	W1	W2	W3	W4	W5	W6	W7		
									Y$

The blank spaces in the table are the variables which are not used at all in your program. (If you want the table sent to a printer, see Operating Hints).

Press the PAGE key and the actual cross-reference will be printed:

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

VARIABLES:

```
AØ
 3Ø  4Ø5Ø

A6
 3Ø  422Ø    3Ø  422Ø    3Ø  422Ø    3Ø  422Ø    3Ø  426Ø *  3Ø  427Ø

A9
 29  4Ø1Ø    3Ø  4Ø1Ø    3Ø  428Ø *  3Ø  428Ø    3Ø  43ØØ *  3Ø  43ØØ
 3Ø  432Ø *  3Ø  432Ø

B$
 29  484Ø *  29  487Ø *  29  499Ø    29  511Ø    29  514Ø    29  635Ø *
 29  636Ø

DØ
 29  592Ø    29  594Ø    29  6Ø7Ø *

D3
 29  416Ø *

D$
 3Ø  414Ø

EØ
 29  438Ø *

E1
 3Ø  423Ø *

F1
 29  4Ø7Ø *  29  4Ø8Ø    29  4Ø9Ø    29  41ØØ    29  411Ø    29  415Ø
 29  432Ø    29  433Ø    29  447Ø    29  461Ø    29  462Ø    29  471Ø
```

The number 29 and 30 are the tape file numbers in which this line of code appears. For example:

AØ
3Ø 4Ø5Ø means aØ is used in line 4050 of file 30

D3
29 4160 * means D3 is assigned a value (indicated by the asterisk) in line 4160 of file 29.

If your program is on only one tape you may not want the file number printed. This can be done by changing the 1360 REM.... to 1360 F$ = " ".

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

Variables

A (26,12) - an array of "pointers" into N$ - see Methodology

A$ (1) - the character previous to the one being analyzed

B$ (1) - the character being analyzed

C$ (1) - the character following the one in B$

F - the number of the tape file currently being read

FØ = MEMORY of your machine - 250. N$ is considered to be full when it has FØ characters in it.

F1 - beginning file mumber

F2 - ending file number

F3 - scratch used when printing the cross-reference map

F4 - the number of items to print per line (each require 12 spaces)

F5 - the number of lines to print before doing a form feed (control L) - used only if you are using a printer.

F6 = F5 - 2

F7 - scratch used when printing the cross-reference map

F$ (3) - the current tape file number

6$ (5) - the line number of the current line of code

H$ (8) = F$ & 6$

I$ (12) = " Ø 1 2 3 4 5 6 7 8 9 $ "

J - character position in L$ of A$
JØ - character position in L$ of B$
J1 - character position in L$ of C$

J2,J3,J4 - scratch variables

J5 - device address of where the cross-reference information will be printed. See Operating Hints

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

J6 = 2 ↑ 16, used for unpacking the pointers from A

J7 = 1 + 1/J6, used for packing the pointers into A

J8 - position in L$ of the second character after the first space.

J9 = 15, the total number of characters required in N$ for each occurence of every variable

J$ (7) - Scratch

K$ (3) - the first 3 characters of a line of code which follow the first space. (eg. K$ might be REM or PRI, etc.)

L$ (75) - string to hold a line of input from the file

N - the number of characters currently in N$

N$ - holds the information about each occurrence of every variable. DIM N$ (FØ + 2 * J9), FØ + MEM - 250

R$ (6) - scratch

S$ (10) - 10 spaces, used for padding other strings with spaces

T$ (J9) - holds the information which is to be appended at the end of N$

V$ (2) - the name of a variable (eg A1)

W$ (3) - scratch

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

```
1000 REM  VARIABLE CROSS-REFN.,  DAN TAYLOR, APRIL 1976, WILSONVILLE
1010 REM  CHANGE: F4=10, F5=60, J5=41   IF YOU WANT TO USE A PRINTER

1020 INIT
1030 F4=6
1040 F5=1.0E+300
1050 J5=32
1060 DIM A(26,12),F$(3),G$(5),H$(8),I$(12),J$(7),R$(6),S$(10)
1070 J9=15
1080 DIM A$(1),B$(1),C$(1),K$(3),L$(75),T$(J9),V$(2),W$(3)
1090 F0=MEMORY-250
1100 DIM N$(F0+2*J9)
1110 I$=" 0123456789$"
1120 S$="           "

1130 REM ----- GET FILE NUMBER

1140 PRINT "LCROSS-REFERENCE AND LIST PROGRAM VARIABLES"
1150 PRINT "JNOTE: YOUR PROGRAM MAY BE ON SEVERAL TAPE FILES.GGGG"
1160 PRINT "JBegin with File Number = ";
1170 INPUT F1
1180 IF INT(F1)<>F1 OR F1<=0 THEN 1160
1190 PRINT "  JEnd with File Number = ";
1200 INPUT F2
1210 IF INT(F2)<>F2 OR F2<=0 THEN 1190

1220 REM ----- INITIALIZE

1230 N=1
1240 N$=""
1250 J7=1+1/2↑16
1260 A=0
1270 SET NOCASE
1280 PRINT "J"
1290 FOR F=F1 TO F2
```

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

```
1300      PRINT "JFILE # ";F
1310      FIND F
1320      J$=STR(F)
1330      J$=REP("",1,1)
1340      R$=SEG(S$,1,3-LEN(J$))
1350      F$=R$&J$

1360      REM SET F$="   " (3 SPACES) IF YOU DON'T WANT THE FILE NUMBER

1370      ON EOF (0) THEN 1390
1380      GO TO 1970
1390 NEXT F

1400 REM ----- PRINT NUMERIC VARIABLE NAMES

1410 PRINT @J5:"L";
1420 PRINT @J5:"VARIABLES:J"
1430 FOR J3=1 TO 26
1440      V$=CHR(64+J3)
1450      FOR J4=1 TO 12
1460           IF A(J3,J4) THEN 1490
1470           PRINT @J5:"      ";
1480           GO TO 1510
1490           W$=SEG(I$,J4,1)
1500           PRINT @J5:V$;W$;"    ";
1510      NEXT J4
1520      PRINT @J5:
1530 NEXT J3
1540 IF J5<>32 THEN 1560
1550 PRINT "GGGGGGGG↑KJJ"
1560 PRINT @J5:"LVARIABLES:J"
1570 J6=1/(J7-1)
1580 F6=F5-2
1590 F7=2
1600 FOR J3=1 TO 26
1610      V$=CHR(64+J3)
```

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

```
1620     FOR J4=1 TO 12
1630         IF NOT(A(J3,J4)) THEN 1920
1640         W$=SEG(I$,J4,1)
1650         IF F7<F6 THEN 1680
1660         PRINT @J5:"L";
1670         F7=0
1680         PRINT @J5:"J";V$;W$
1690         F7=F7+2
1700         F3=0
1710         J2=J6*(A(J3,J4)-INT(A(J3,J4)))
1720         W$=SEG(N$,J2,3)
1730         T$=SEG(N$,J2+3,5)
1740         C$=SEG(N$,J2+8,1)
1750         F3=F3+1
1760         IF F3<=F4 THEN 1830
1770         F3=1
1780         PRINT @J5:
1790         F7=F7+1
1800         IF F7<F5 THEN 1830
1810         PRINT @J5:"L";
1820         F7=0
1830         PRINT @J5:W$;" ";T$;" ";C$;" ";
1840         T$=SEG(N$,J2+9,6)
1850         J2=VAL(T$)
1860         IF J2 THEN 1720
1870         PRINT @J5:
1880         F7=F7+1
1890         IF F7<F5 THEN 1920
1900         PRINT @J5:"L";
1910         F7=0
1920     NEXT J4
1930 NEXT J3
1940 PRINT @J5:
1950 RETURN

1960 REM ----- SUBROUTINE TO FETCH VARIABLE NAMES FROM INPUT LINE
```

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

```
1970 INPUT @33:L$
1980 J=POS(L$," ",2)
1990 J0=J+1
2000 K$=SEG(L$,J0,3)
2010 IF K$="REM" OR K$="IMA" THEN 1970
2020 J8=J0+1
2030 G$=SEG(L$,1,J-1)
2040 R$=SEG(S$,1,5-LEN(G$))
2050 G$=R$&G$
2060 H$=F$&G$
2070 L$=L$&"    "
2080 A$=SEG(L$,J,1)
2090 B$=SEG(L$,J0,1)
2100 FOR J1=J8 TO LEN(L$)-2

2110     REM ----- IGNORE CHARACTERS BETWEEN QUOTES

2120     IF B$<>"""" THEN 2200
2130     J=POS(L$,"""",J1)
2140     J0=J+1
2150     J1=J0+1
2160     A$=SEG(L$,J,1)
2170     B$=SEG(L$,J0,1)
2180     GO TO 2120

2190     REM ----- CHECK PREVIOUS(A$) AND SUCCEEDING(C$) CHARACTER

2200     C$=SEG(L$,J1,1)
2210     IF B$<"A" OR B$>"Z" THEN 2860
2220     IF C$>"@" AND C$<"[" THEN 2790
2230     IF A$>"@" AND A$<"[" THEN 2860
2240     IF A$>"/" AND A$<":" THEN 2860
2250     J3=ASC(B$)-64

2260     REM ----- ONE OR TWO CHARACTER NAME--SKIP ONE OR TWO CHARS.
```

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

```
2270      IF C$="$" OR (C$>"/" AND C$<":") THEN 2360
2280      V$=B$&" "
2290      A$=C$
2300      J=J1
2310      J1=J1+1
2320      J0=J1
2330      B$=SEG(L$,J0,1)
2340      J4=1
2350      GO TO 2440
2360      J4=POS(I$,C$,2)
2370      V$=B$&C$
2380      J=J1+1
2390      A$=SEG(L$,J,1)
2400      J0=J+1
2410      B$=SEG(L$,J0,1)
2420      J1=J0

2430      REM ----- ASSIGNMENT ?

2440      IF K$="INP" OR K$="REA" OR K$="RBY" THEN 2500
2450      W$=" "
2460      IF A$<>"=" THEN 2520
2470      IF K$="IF " THEN 2520
2480      IF POS(L$,"=",J8)<J THEN 2520
2490      GO TO 2510
2500      IF POS(L$,":",8)>J0 THEN 2520
2510      W$="*"
2520      J6=A(J3,J4)
2530      IF J6 THEN 2580

2540      REM ----- FIRST OCCURRENCE OF A(J3,J4)

2550      A(J3,J4)=N*J7
2560      GO TO 2630
```

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

```
2570     REM ----- ANOTHER OCCURRENCE

2580     J$=STR(N)
2590     R$=SEG(S$,1,6-LEN(J$))
2600     J$=R$&J$
2610     N$=REP(J$,J6+9,6)
2620     A(J3,J4)=J6-INT(J6)+N
2630     J$=W$&"-0END-"
2640     T$=H$&J$
2650     N$=REP(T$,N,0)
2660     N=N+J9
2670     IF N<F0 THEN 2900
2680     PRINT @J5:"LSTORAGE FILLED IN THE MIDDLE OF FILE # ";F;".  ";
2690     PRINT @J5:"THE PROGRAM WILL"
2700     PRINT @J5:"CONTINUE WITH FILE # ";F;" AFTER ";
2710     PRINT @J5:"THE FOLLOWING INFORMATION IS PRINTED:JJ"
2720     GOSUB 1420
2730     A=0
2740     N=1
2750     N$=""
2760     PRINT @J5:"JCONTINUING WITH FILE # ";F
2770     GO TO 2900

2780     REM ----- SKIP NEXT CHARACTER

2790     A$=C$
2800     J=J1
2810     J1=J1+1
2820     J0=J1
2830     B$=SEG(L$,J0,1)
2840     GO TO 2900

2850     REM ----- PREPARE FOR NEXT CHARACTER

2860     J=J0
2870     J0=J1
```

TITLE	ABSTRACT NO:
CROSS-REFERENCE & LIST PROGRAM VARIABLES	51/00-8004/0

```
2880      A$=B$
2890      B$=C$
2900 NEXT J1
2910 GO TO 1970
```

Tape 19 Files 1-7

TEKTRONIX

4051

APPLICATIONS LIBRARY PROGRAM

TITLE		ABSTRACT NUMBER
Gantt Chart		51/07-0102/0
ORIGINAL DATE February 1980	REVISION DATE	MEMORY REQUIREMENT 32K
AUTHOR Connie Breithaupt	Tektronix, Inc. Rockville, MD	PERIPHERALS See Below

ABSTRACT

Peripherals: 4907 File Manager (For Disk Version Only)
Optional - 4662 Plotter

Statements: Tape Version - 1617
Disk Version - 1605

Files: Tape Version - 4 ASCII Program
3 Binary Data (Directory and Example)
Requires pre-MARKed data files

Disk Version - 4 Binary Program
10 Binary Data (Examples)

The program generates a chart and graph of activity vs. time.

The following is input:

Project name - 2 lines - 15 characters each
Chart title - 45 characters
Code - 16 characters
Project code - 6 characters
Signatures - 2 lines - 30 characters each
Vertical row descriptions - single or double spaced - 4 rows maximum
Horizontal row descriptions - single or double spaced - 16 rows maximum
Weekly data for each row - 28 weeks maximum - beginning with a Sunday

Charts are saved by name. File 5 of the tape version contains a directory of the charts by name.

Weekly data may be edited for each week or selected weeks.

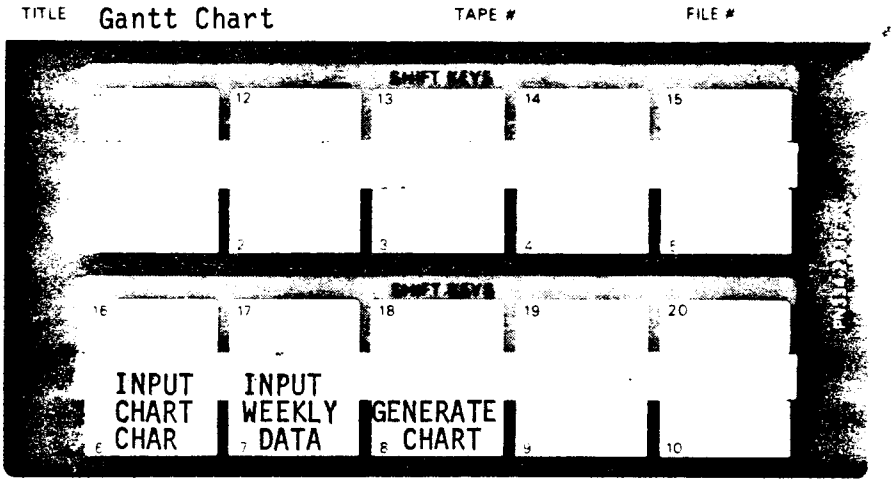
Charts are generated for a 14 week period starting with a chosen beginning column date.

Three selected rows of data may be graphed on one chart. The first selected row will be graphed with a solid line, the second with a dotted line, and the third with a dashed line.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NUMBER
Gantt Chart	51/07-0102/0

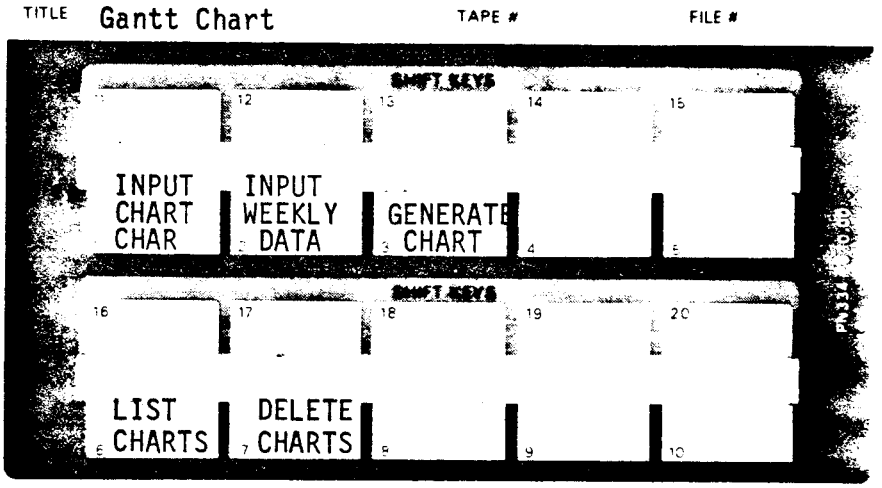
Disk Version



UDK #	Function
6	Input chart characteristics
7	Input weekly data
8	Generate the chart on the screen or the plotter

TITLE	ABSTRACT NUMBER
Gantt Chart	51/07-0102/0

Tape Version



UDK #	Function
1	Input chart characteristics
2	Input weekly data
3	Generate the chart on the screen or the plotter
6	List the existing chart names and file numbers
7	Delete all charts

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

LOADING INSTRUCTIONS

Tape Version

Press AUTOLOAD. The program should be located on the first files of the tape or will need to be recoded.

File 1	Autoload
File 2	Describe chart characteristics
File 3	Input weekly data
File 4	Generate chart
File 5	Chart names & storage file numbers
File 6	Chart #1 - chart characteristics (923 byte string arry - A$)
File 7	Chart #1 - weekly data (beginning data date (R$), number of data rows (L1) and the actual weekly data (D9(L1,28)))

Disk Version

Mount the disk and enter:

```
OLD "@GANTT1/AUTOLOAD"
RUN
```

The program consists of the following files:

@GANTT1/AUTOLOAD	Autoload
@GANTT1/PROGRAM/INPUT/CHART	Describe characteristics
@GANTT1/PROGRAM/INPUT/DATA	Input weekly data
@GANTT1/PROGRAM/OUTPUT/NODOC	Generate chart
@GANTT1/MODEL/CHART	Examples
@GANTT1/MODEL/DATA	
@GANTT1/JETSTAR1/CHART	
@GANTT1/JETSTAR1/DATA	
@GANTT1/JETSTAR2/CHART	
@GANTT1/JETSTAR2/DATA	
@GANTT1/ALLPROJECT/CHART	
@GANTT1/ALLPROJECT/DATA	
@GANTT1/INTERCO1/CHART	
@GANTT1/INTERCO1/DATA	

@GANTT1/chartname/CHART
1000 byte, sequential

(923 byte string array containing chart characteristics)

@GANTT1/chartname/DATA
1000 byte, sequential

(date in string format, number of rows, numeric array dimensioned to number of rows x 28 containing weekly data)

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

Tape Version

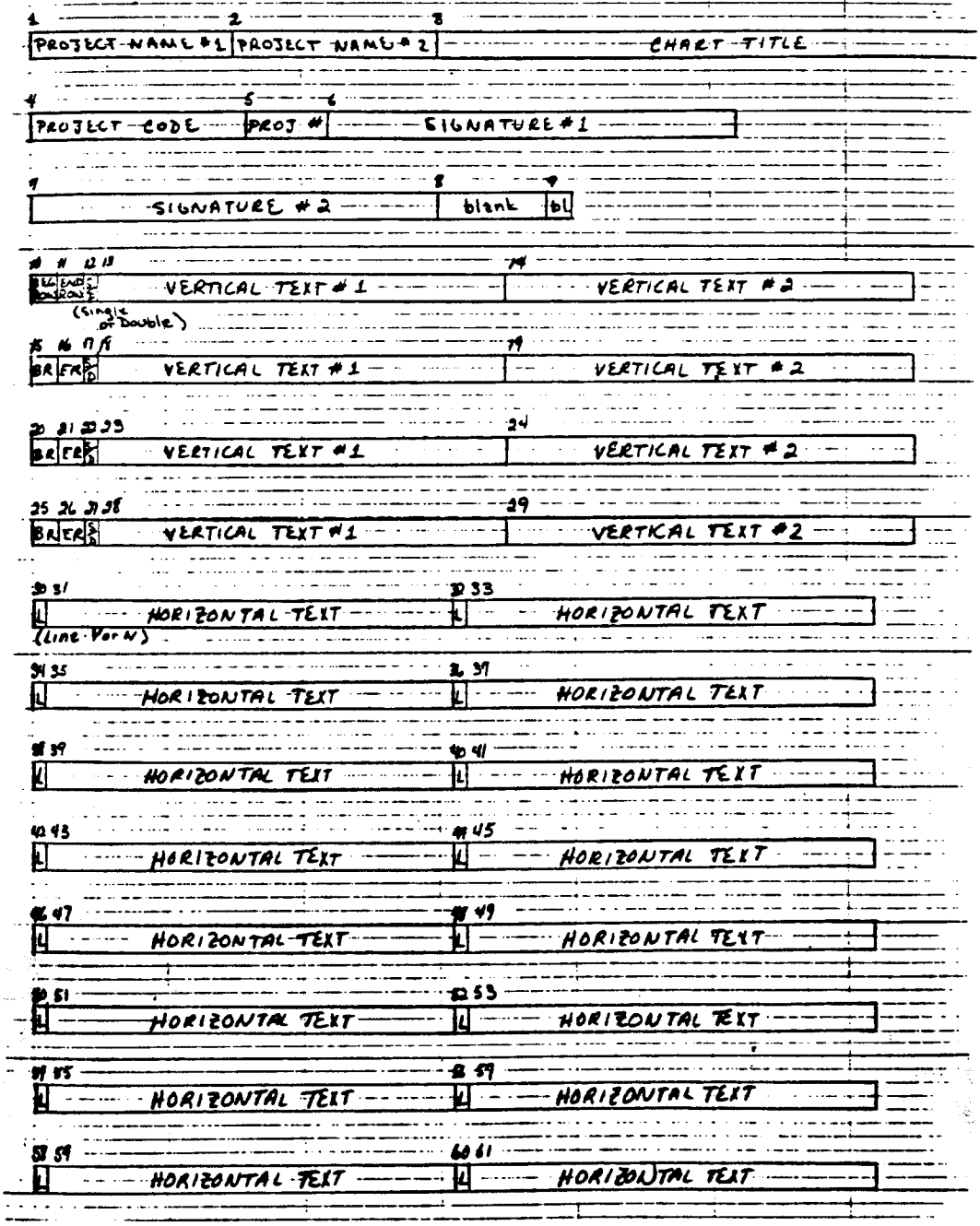
Directory File #5 and data files may be created in the following manner.

```
FIND 5
MARK 1,3072
FIND 5
WRITE @33:"DONE",6
```

Each chart requires two data files one for the characteristics and one for the data.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

This is the layout of A$(923) which contains the chart characteristics. Array A1(61) contains the indexes to each element of A$. Ex: A1(3)=31 is the beginning position of CHART TITLE.



TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

OPERATING INSTRUCTIONS

Tape Version

* GANTT CHARTS *

UDK	TASK
1	DESCRIBE CHART
2	INPUT WEEKLY DATA
3	GENERATE CHART
6	LIST STORED CHART NAMES
7	INITIALIZE DATA FILES (DELETE ALL CHARTS)

Press a Function Key!

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

Disk Version

```
* GANTT CHARTS *


     UDK TASK
     --- ----

         *TAPE VERSION*

      1   DESCRIBE CHART
      2   INPUT WEEKLY DATA
      3   GENERATE CHART

         *DISK VERSION*

      6   DESCRIBE CHART
      7   INPUT WEEKLY DATA
      8   GENERATE CHART


     The 'Generate Chart' Task will not return
      to the Menu (Memory Considerations)!

     Press a Function Key!
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

This Task allows characteristics of a GANTT Chart to be described.

```
  PROJECT NAME: JEISIBB_II_____
   CHART TITLE: SHORTAGE_SUMMARY_____________________________
          CODE:
PROJECT NUMBER: __7005
     SIGNATURE: PREPARED_BY:_H.V._HENDERSON___
                RESP:_V.H._BRADY/J.M._ANDERSON
```

A$(923) is initialized to Blanks. As the Chart characteristics are input they are stored in A$ at positions defined by A1(61). For example A1(3)=31 is the First Position of CHART TITLE.

A one line PROJECT NAME must be typed on the first indicated line. The second line is optional.

The CHART TITLE will be centered by the program. This is the only text automatically centered.

The PROJECT CODE is optional.

A one line signature must be typed on the first indicated line. The second one is optional.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
VERTICAL DESCRIPTIONS (4 MAX)


  SINGLE or DOUBLE: D (type '0' when done)
           BEG ROW: 1_
           END ROW: 4_
    DESCRIPTION #1: ___CANCEL__
    DESCRIPTION #2: ____JOBS___


  SINGLE or DOUBLE: S (type '0' when done)
           BEG ROW: 5_
           END ROW: 6_
       DESCRIPTION: _CUM_


  SINGLE or DOUBLE: 0 (type '0' when done)
```

The VERTICAL ROW DESCRIPTION's can be 1 line or 2 lines (Single or Double). The END ROW of the description must be greater than the BEG ROW. Remember, there is a maximum of 16 Rows. The indicated rows must not already be spanned by another VERTICAL DESCRIPTION. Rows spanned by single descriptions are stored in A2(16), those by double descriptions in A3(16); these arrays are checked when new VERTICAL DESCRIPTION's are defined to prevent overlapping.

The number of rows spanned by VERTICAL DESCRIPTION determines the number of spaces available for the actual text. In this program the rows are 3 units and the text print 1 unit in width. The formula then is:

(# of spanned rows) x 2 + (# of spanned rows - 1)

Ex: spanned rows are 3 thru 6 thus

[(4) x 2] + [(4) - 1] = 11 possible characters

The text for each VERTICAL DESCRIPTION is only centered only if it is centered at the time of input. See above example.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

HORIZONTAL DESCRIPTIONS

DESCRIPTION: MAJOR_ASSEMBLY----------
type '0' when complete
LINE? N

DESCRIPTION: SUBASSEMBLY----------
type '0' when complete
LINE? N

DESCRIPTION: FEEDER_PLANTS----------
type '0' when complete
LINE? N

DESCRIPTION: __TOTAL----------
type '0' when complete
LINE? Y

DESCRIPTION: MAJOR_ASSEMBLY----------
type '0' when complete
LINE? N

DESCRIPTION: SUBASSEMBLY----------
type '0' when complete
LINE? N

DESCRIPTION: FEEDER_PLANTS----------
type '0' when complete
LINE? N

DESCRIPTION: __TOTAL----------
type '0' when complete
LINE? Y

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
DESCRIPTION: TOTAL_LINE_&_CUSHION__________
             type '0' when complete
       LINE? N

DESCRIPTION: 0_____________________________
             type '0' when complete
```

The DESCRIPTION's will be printed on the chart exactly as indicated, e.g., in the example on the preceding page, TOTAL will be indented 2 spaces.

The number of characters allowed for the DESCRIPTION (L3) is 30 if there is not a vertical description on the row, 27 if the vertical description is single, 25 if the vertical description is double.

A "Y" in answer to LINE? will generate a horizontal line following the DESCRIPTION. Note that a LINE should not be indicated after the last DESCRIPTION as the program generates the bottom outline.

A line should also be indicated after the row which completes a VERTICAL DESCRIPTION. When the VERTICAL DESCRIPTION is printed only the vertical line is drawn.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

HORIZONTAL DESCRIPTIONS

```
DESCRIPTION: GELAC__________SCHEDULE__
             type '0' when complete
       LINE? N

DESCRIPTION: DCC____________ACTUAL____
             type '0' when complete
       LINE? N

DESCRIPTION: REL____________CUM_______
             type '0' when complete
       LINE? N

DESCRIPTION: _____________UNREL-BS__
             type '0' when complete
       LINE? Y

DESCRIPTION: GELAC__________ACTUAL____
             type '0' when complete
       LINE? N

DESCRIPTION: LIAISON________CUM_______
             type '0' when complete
       LINE? 0
```

Two column HORIZONTAL DESCRIPTIONS can be created. Type the two descriptions separated by enough spaces to later draw the dividing line by hand.

In this example, notice that the words for the second column are typed 10 spaces from the right.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

Tape Version

```
What shall be the name of this Chart? JEISI8R_II
```

Disk Version

```
What shall be the name of this Chart? JEISI8R_II

Insert in Unit #0 the Disk to contain Chart JETSTAR II
To continue, type <CR>.
```

The chart name must be 10 Alphanumeric characters or less and begin with an Alphabetic character.

If the indicated chart name already is stored in File #5 it probably means this task was used to modify the chart characteristics. In case it was a mistake, though, a message will print, offering the choice of overwritting the already stored chart or not. A "Y" will overwrite the stored chart; an "N" will allow a new chart name to be indicated. Likewise, a message will print, offering the choice of overwritting the chart's data file or not. A "Y" will keep the existing data; an "N" will initialize the data file.

```
JETSTAR II Chart format stored!
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
This Task allows Data Editing.

Which Chart? JEISI8R_II

Beginning Data Date is 01/01/50.  Is this correct? N

New Date: 9/3/78__
```

The chart name will be rejected unless the first character is Alphabetic and the remaining 9 are Alphanumeric.

The program will check if the indicated chart is indeed stored on the tape.

The data file is read: R$ - beginning data date, L1 - number of rows, D9(L1,42) - data.

The beginning data date is the date associated with the first column of data. Notice that zeroes do not have to be typed for the 1 digit month or day. If the beginning data date is modified, it must be a Sunday. It is tored in P$, as R$ is the variable which must be passed to the calendar routines.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

During Data modification, type:

<CR> if present value is correct;
-1 if value is not to print; or
0 if it is to be calculated.

Typing a '0' for any of the following Questions will cause an exit from the indicated Process.

Present row value is retained, nothing will print when chart is generated.

A calculation will override any existing row value, so a "0" or actually any value can be typed.

Do you want to go thru each Week(1), only selected Weeks(2), or Neither(0)? 1

Should Data for 9/3/78 be Modified? Y

Row 1:	14/04	------
Row 2:	7	8-----
Row 3:	6	7-----
Row 4:	27/04	29/04_
Row 5:	1/02	------
Row 6:	1/02	------
Row 7:	2	------
Row 8:	4/04	------
Row 9:	31/08	33/08_

Should Data for 9/10/78 be Modified? N

If modification Type 1 is selected, each weekly date can be modified in order, starting form the beginning data date.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

During Data modification, type:

<CR> if present value is correct;
-1 if value is not to print; or
0 if it is to be calculated.

Typing a '0' for any of the following Questions will cause an exit from the indicated Process.

Do you want to go thru each Week(1), only selected Weeks(2), or Neither(0)? 2

Which Date: 9/17/78_
Row 1: 26/22 ------
Row 2: 6 ------
Row 3: 5 ------
Row 4: 37/22 ------
Row 5: 2/01 ------
Row 6: 5/01 2/01__
Row 7: 0/01 ------
Row 8: 4/03 ------
Row 9: 41/25 ------

Which Date: --------

Modification Type 2 allows the operator to indicate the date to be modified.

When a data value is typed, a slash is changed to a decimal point and stored in a numeric array (D9). Before printing the value back on the screen or plotter, the "." is changed back to a "/". This method used to facilitate the total calculations (next page). Because this method was used, a value like 39/2 must be typed as 39/Ø2 (otherwise it will be stored and used as 39/2Ø).

Also, if the value was input with a slash (changed to a decimal point) .0001 is added to it; so a value like 24/0 is stored as 24.0001, not 24. (the last two digits, 01, will be dropped before printing).

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

Total Calculations

How many Rows? 3_ (type '0' when complete)

Indicate Rows: 1_2_3_____

Indicate Total Row #: 4_

Total Calculations

How many Rows? 3_ (type '0' when complete)

Indicate Rows: 5_6_7_____

Indicate Total Row #: 8_

Total Calculations

How many Rows? 2_ (type '0' when complete)

Indicate Rows: 4_8_____

Indicate Total Row #: 9_

Totals are found across all 28 columns of the indicated rows. Addition is the only function available.

Rows to be added can not be the same as the total row.

Rows which were defined by the total calculation can later be used as rows to be added. See above example.

If any column of the rows to be added is -1, that column of the total row will also be -1, i.e., nothing will print in the indicated row/column when the chart is drawn.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

Would you like a Print-out of the existing Data? N

Would you like to go thru the Editing process again? N

Beginning Date, Number of Rows, and the Weekly Data has been stored in the Data File for Chart JETSTAR II!

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
This Task generates GANTT Chart.

What is the Name of the GANTT Chart? JEISI88_II

Beginning Data Date is 9/3/78
Indicate beginning Column Date: 9/17/78_
```

The program will print the date of the first column of data. The operator then indicates the data of the first column to be charted. This date must be a Sunday and be within 28 of the beginning date date. Note that a "Ø" does not have to be typed before a 1 character month or day.

The program will chart 14 weeks of data starting with the indicated beginning column date. This task can be run again, with a different beginning column date.

The following examples show this task run on the same chart with starting dates 9/17/78 and 12/10/78.

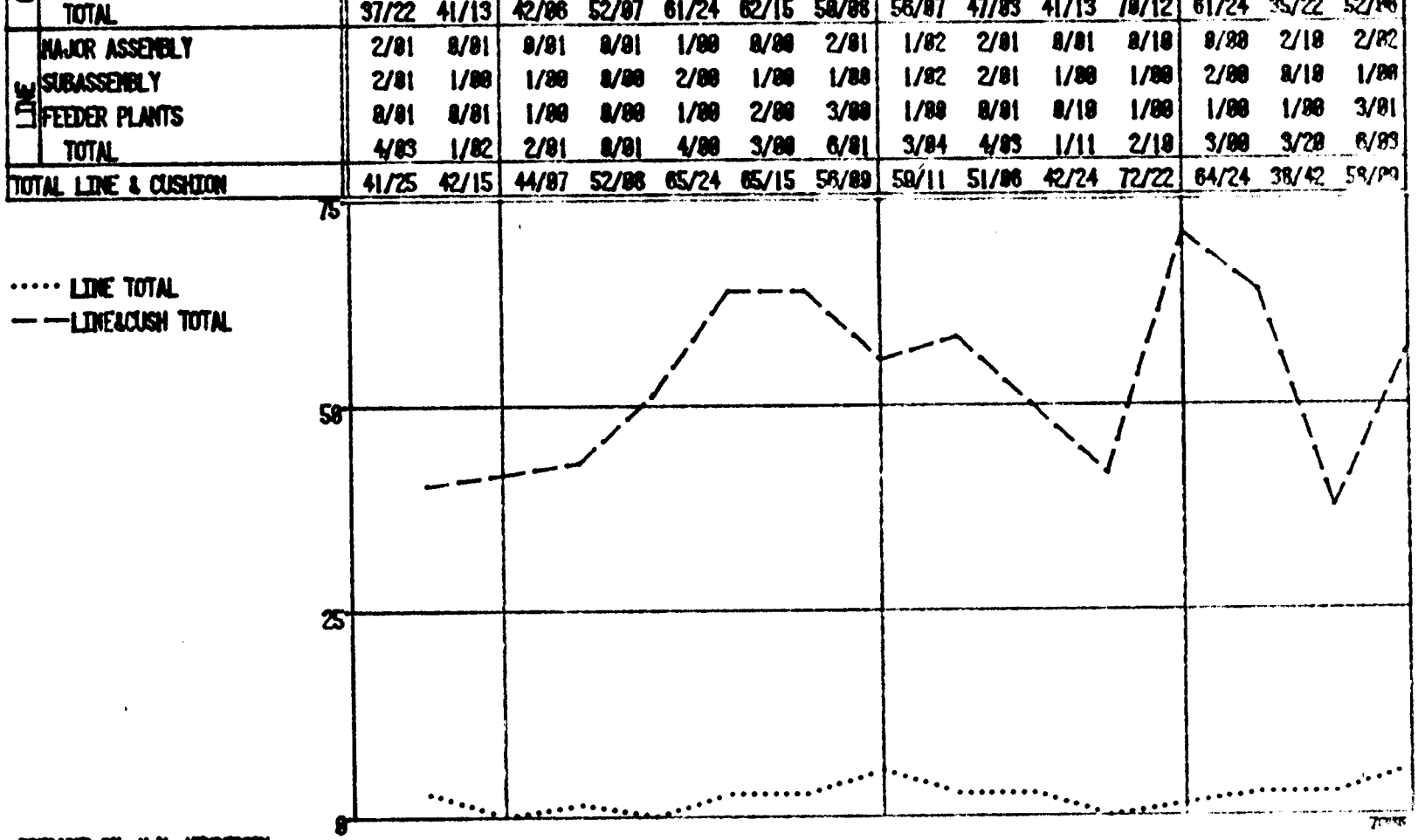
Remember that the data file holds 28 weeks of data. When the beginning column date is within 14 weeks of the end, the program widens the width of the columns to compensate for the lesser number of columns. See the third example which follows.

000-6405-02

JET STAR II

SHORTAGE SUMMARY

	ACTIVITY	1978 SEP 17	SEP 24	OCT 1	OCT 8	OCT 15	OCT 22	OCT 29	NOV 5	NOV 12	NOV 19	NOV 26	DEC 3	DEC 10	DEC 17
CUSHION	MAJOR ASSEMBLY	26/22	32/13	31/06	36/04	48/21	36/14	29/06	38/07	36/03	32/13	58/02	48/21	26/22	36/04
	SUBASSEMBLY	6/00	9/00	9/00	7/00	12/00	14/00	13/00	12/00	6/00	9/00	28/10	12/00	6/00	7/00
	FEEDER PLANTS	5/00	8/00	2/00	9/03	9/03	12/01	8/00	6/00	5/00	8/00	8/00	9/03	3/00	9/02
	TOTAL	37/22	41/13	42/06	52/07	61/24	62/15	50/06	56/07	47/03	41/13	78/12	61/24	35/22	52/06
LINE	MAJOR ASSEMBLY	2/01	0/01	0/01	0/01	1/00	0/00	2/01	1/02	2/01	0/01	0/10	0/00	2/10	2/02
	SUBASSEMBLY	2/01	1/00	1/00	0/00	2/00	1/00	1/00	1/02	2/01	1/00	1/00	2/00	0/10	1/00
	FEEDER PLANTS	0/01	0/01	1/00	0/00	1/00	2/00	3/00	1/00	0/01	0/10	1/00	1/00	1/00	3/01
	TOTAL	4/03	1/02	2/01	0/01	4/00	3/00	6/01	3/04	4/03	1/11	2/10	3/00	3/20	6/03
TOTAL LINE & CUSHION		41/25	42/15	44/07	52/08	65/24	65/15	56/09	59/11	51/06	42/24	72/22	64/24	38/42	58/09



PREPARED BY: N.V. HENDERSON
REP: V.M. BRADY/J.M. ANDERSON

TITLE
Gantt Chart

ABSTRACT NO:
51/07-0102/0

PAGE NO: 20

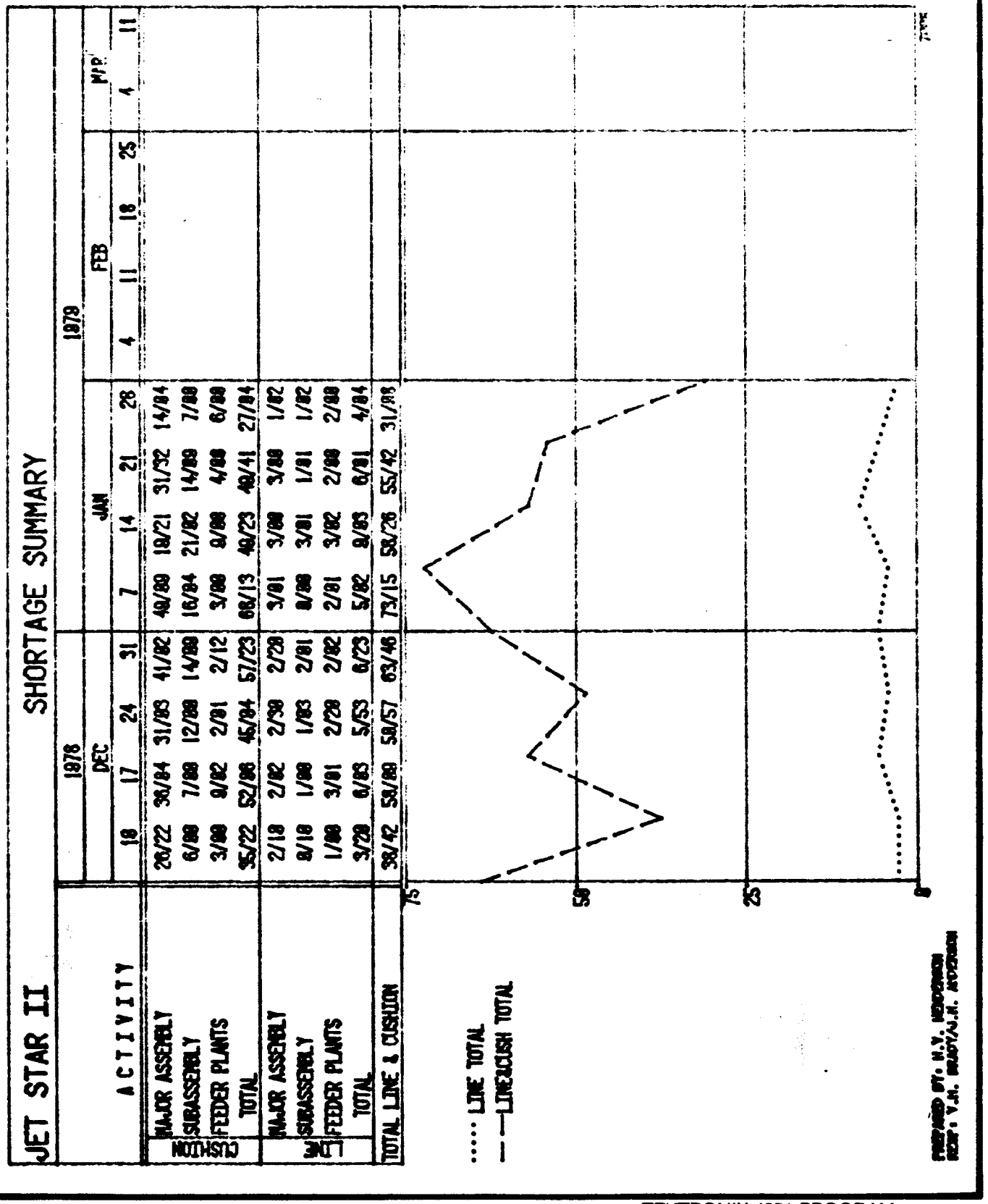
TEKTRONIX 4051 PROGRAM

TITLE

Gantt Chart

ABSTRACT NO:

51/07-0102/0



000-6405-02

TEKTRONIX 4051 PROGRAM

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

Row Descriptions for Chart JETSTAR II

ROW 1: MAJOR ASSEMBLY
ROW 2: SUBASSEMBLY
ROW 3: FEEDER PLANTS
ROW 4: TOTAL
ROW 5: MAJOR ASSEMBLY
ROW 6: SUBASSEMBLY
ROW 7: FEEDER PLANTS
ROW 8: TOTAL
ROW 9: TOTAL LINE & CUSHION

Indicate the 3 Rows to be Graphed: 0_8_9___
(If necessary, type 0's to make 3)

To help select the rows which are to be graphed, the row descriptions are listed.

The numbers of 3 rows must be indicated for graphing. If necessary, type "0"s.

The first selected row will be graphed with a solid line, the second with a dotted line, and the third with a dashed line.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
Legend name for:

  Row 8: LINE_TOTAL_____
  Row 9: LINE&CUSH_TOTAL

The minimum Y value of the selected Rows is 0
and the maximum Y value is 72

Indicate graph's:
  Minimum Y value: 0_____
  Maximum Y value: 100___
  Interval       : 25____

Shall the Chart be generated on the Screen(32) or on
the Plotter(1)? 1_

Ready Plotter!
To continue, type <CR>.
```

The operator must indicated the legend names of each row to be graphed (O$, P$, Q$).

The program will print the minimum and maximum Y values of the range indicated by the 3 rows to be graphed. The operator then indicates the graph range (H5 & H6) and grid line interval (H7). The interval must evenly divide the range.

If the chart is to be generated on the plotter ("1" or the plotter's device number) the program will pause so the plotter can be readied. The output device variable is D1.

If the chart is generated on the screen, the Alphanumeric characters will be software generated. Because of the number of characters required for the Gantt Chart, hardware characters cannot be used. Characters other than alphanumeric will be displayed as a period ".". The subroutine which strokes the screen characters does not handle angled text, thus the vertical descriptions cannot be printed on the screen. Variable S contains all the character strokes.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

If the program indicates that both the minimum and maximum Y values are "Ø", this means that no values have been typed for the indicated column range, i.e., data values are all -1. The operator should still type the range values and interval; the graph will be drawn but without curves.

If no rows have been chosen to be graphed; i.e., all "Ø's" were typed, the operator should still select the graph's range and interval; the graph will be drawn without curves.

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

MODIFICATIONS

It will be simple to modify the CHART GENERATION program to chart monthly, instead of weekly data.

Some of the changes:

Change G7 = 3 to G7 = Ø. This is presently the width of the week area.

In the Calendar Subroutine, change T1=2 (indicating weeks) to T1 = (indicating months).

Skip lines of code which presently print out the week date. Also, skip the code which checks if the month has changed; a line should be drawn after all columns.

Miscellaneous text can also be added to a chart. After the chart is plotted, move the plotter pen to where the additional text is to be printed and type:

```
PRINT @1,17:char width, char height
PRINT @1:"text"
```

Be careful when renumbering or modifying the autoload and output program. The output program deletes line numbers and these may have to be modified.

000-6405-02

PROJECT NAME #1
PROJECT NAME #2

CHART TITLE (centered)

PROJECT CODE

ACTIVITY	1950 APR 9	16	23	30	MAY 7	14	21	28	JUN 4	11	18	25	JUL 2	9
DOUBLE VERT DESCRIPTION: HORIZONTAL DESCRIPTION NEXT TO A DOUBLE VERT DESCRIPTION GETS 25 SPACES FOR TEXT														
SINGLE VERT DESCRIP: HORIZONTAL DESCRIPTION NEXT TO A SINGLE VERTICAL DESCRIPTION GETS 27 SPACES FOR TEXT TEXT CAN BE INDENTED FOR SPECIAL REASONS														
HORIZONTAL DESCRIPTIONS WITH NO VERT DESC's GET 30 SPACES FOR TEXT														
Lines can be drawn under Rows														
A MAXIMUM OF 16 ROWS														

100
75
50
25
0

PROJ ◆

SIGNATURE #1
SIGNATURE #2

TITLE
Gantt Chart

ABSTRACT NO:
51/07-0102/0

PAGE NO: 26

TEKTRONIX 4051 PROGRAM

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

TAPE VERSION -- FILE 1

```
0 GO TO 100

3 REM ************* INPUT OF CHART CHARACTERISTICS ***** TAPE VERSION

4 FIND F1*3-1
5 GOSUB 540
6 RETURN

7 REM ************************ INPUT OF WEEKLY DATA ***** TAPE VERSION

8 FIND F1*3
9 GOSUB 540
10 RETURN

11 REM ********************** GENERATE GANTT CHART ***** TAPE VERSION

12 FIND F1*3+1
13 OLD
14 RETURN

23 REM ********************************** LIST NAMES OF STORED CHARTS

24 GOSUB 400
25 GOSUB 270
26 RETURN

27 REM ************************************** DELETE ALL STORED CHARTS

28 GOSUB 510
29 GOSUB 380
30 RETURN
100 INIT
110 PRINT "L* GANTT CHARTS *JJ"
260 F1=1
270 PRINT "JJJ     UDK  TASK"
280 PRINT "     ---  ----"
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
290 PRINT "J       1   DESCRIBE CHART"
300 PRINT "       2   INPUT WEEKLY DATA"
310 PRINT "       3   GENERATE CHART"
320 PRINT "J      6   LIST STORED CHART NAMES"
330 PRINT "       7   INITIALIZE DATA FILES"
340 PRINT "           (DELETE ALL CHARTS)"
380 PRINT "JJ     Press a Function Key!GG"
390 RETURN
400 FIND 5
410 F1=0
411 PRINT "LEXISTING CHART NAMES & FILE #J"
420 ON EOF (0) THEN 490
430 FOR I1=1 TO 100
440     READ @33:E$,E1
450     IF F1=0 THEN 470
452     PRINT "JJPress <CR> to continue. ";
454     INPUT Z$
455     PRINT "L"
460     RETURN
470     IF E$<>"DONE" THEN 475
472     E$="LAST"
475     PRINT "     ";E$,"        ";E1
480 NEXT I1
490 F1=1
500 RETURN
510 FIND 5
520 WRITE @33:"DONE",6
522 PRINT "JJData Files initialized!GGJ"
530 RETURN
540 SET NOKEY
550 APPEND 730
560 GOSUB 730
570 SET KEY
580 DELETE 731,65000
590 FOR I1=1 TO 700
600 NEXT I1
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
610 PRINT "L"
620 GOSUB 260
630 RETURN
730 INIT
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

TAPE VERSION -- FILE 2

```
100 INIT
110 PRINT "LThis Task allows characteristics of a GANTT Chart"
120 PRINT "to be described.JJ"
130 DIM A$(923),A1(61),A2(16),A3(16),Z$(200)

140 REM ********************************** R4 - LENGTH OF DESCRIPTION AREA

150 R4=31.5

160 REM ************ A$ - CHART CHARACTERISTICS, INITIALIZE TO 923 BLANKS

170 A$="                                                              "
180 FOR I1=1 TO 17
190     A$=A$&"                                                       "
200 NEXT I1
210 A$=A$&"                          "

220 REM ****** A1 - POSITIONS IN A$ WHICH START THE CHARACTERISTICS DESC

230 DATA 1,16,31,76,92,98,128,158,166,168,170,172,173,203,233,235,237
240 DATA 238,268,298,300,302,303,333,363,365,367,368,398,428,429,459
250 DATA 460,490,491,521,522,552,553,583,584,614,615,645,646,676,677
260 DATA 707,708,738,739,769,770,900,801,831,832,862,863,893,894
270 RESTORE 230
280 READ A1

290 REM ***** A2 & A3 - ROWS HAVING SINGLE AND DOUBLE VERT DESC'S, INIT

300 FOR I1=1 TO 16
310     A2(I1)=0
320     A3(I1)=0
330 NEXT I1

340 REM ********************** PROMPT FOR CHARACTERISTICS AND STORE IN A$

350 PRINT "  PROJECT NAME: _____ "
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
360 PRINT "                    _____"
370 PRINT "                    KK";
380 INPUT X$
390 PRINT "                    ";
400 INPUT Y$
410 IF LEN(X$)<>0 THEN 450
420 PRINT "                    type Project Name on Line 1 and optionally";
430 PRINT " on Line 2!GG"
440 GO TO 350
450 IF LEN(X$)<=15 AND LEN(Y$)<=15 THEN 480
460 PRINT "                    15 characters maximum for each Line!GG"
470 GO TO 350
480 A$=REP(X$,A1(1),LEN(X$))
490 IF LEN(Y$)=0 THEN 510
500 A$=REP(Y$,A1(2),LEN(Y$))
510 PRINT "   CHART TITLE: _____________________________________________";
520 PRINT "_____"
530 PRINT "                    K";
540 INPUT X$
550 IF LEN(X$)<=45 AND LEN(X$)<>0 THEN 580
560 PRINT "                    45 characters maximum!GG"
570 GO TO 510
580 A$=REP(X$,A1(3),LEN(X$))
590 PRINT "          CODE: _____HHHHHHHHHHHHHHHH";
600 INPUT X$
610 IF LEN(X$)<=16 THEN 640
620 PRINT "                    16 characters maximum!GG"
630 GO TO 590
640 A$=REP(X$,A1(4),LEN(X$))
650 PRINT "PROJECT NUMBER: _____HHHHHH";
660 INPUT X$
670 IF LEN(X$)<=6 AND LEN(X$)<>0 THEN 700
680 PRINT "                    6 characters maximum!GG"
690 GO TO 650
700 A$=REP(X$,A1(5),LEN(X$))
710 PRINT "     SIGNATURE: _____________________________"
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
720 PRINT "                                                  "
730 PRINT "                KK";
740 INPUT X$
750 PRINT "                ";
760 INPUT Y$
770 IF LEN(X$)<>0 THEN 810
780 PRINT "                type Signature on Line 1 and optionally ";
790 PRINT "on Line 2!GG"
800 GO TO 710
810 IF LEN(X$)<=30 AND LEN(Y$)<=30 THEN 840
820 PRINT "                30 characters maximum for each Line!GG"
830 GO TO 710
840 A$=REP(X$,A1(6),LEN(X$))
850 IF LEN(Y$)=0 THEN 870
860 A$=REP(Y$,A1(7),LEN(Y$))
870 FOR I1=1 TO 300
880 NEXT I1
890 L4=5
900 L1=0
910 PRINT "LVERTICAL DESCRIPTIONS (4 MAX)J"
920 L4=L4+5
930 L1=L1+1
940 IF L1>4 THEN 1720
950 PRINT "J"
960 PRINT " SINGLE or DOUBLE: _ (type '0' when done)"
970 PRINT "                   K";
980 INPUT W$
990 IF W$="0" OR W$="S" OR W$="D" THEN 1020
1000 PRINT "                    type '0','S', or 'D'!GG"
1010 GO TO 960
1020 IF W$="0" THEN 1720
1030 PRINT "          BEG ROW: __HH";
1040 INPUT X$
1050 IF LEN(X$)=1 OR LEN(X$)=2 THEN 1080
1060 PRINT "                    type 1 or 2 numbers please!GG"
1070 GO TO 1030
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
1080 GOSUB 2720
1090 IF J1=0 THEN 1120
1100 PRINT "                    type numbers only!GG"
1110 GO TO 1030
1120 J1=VAL(X$)
1130 IF J1=>1 AND J1<=16 THEN 1170
1140 PRINT "                    Vertical Description can span ";
1150 PRINT "Rows 1 thru 16 only!GG"
1160 GO TO 1030
1170 Y$=X$
1180 PRINT "          END ROW:  HH";
1190 INPUT X$
1200 IF LEN(X$)=1 OR LEN(X$)=2 THEN 1230
1210 PRINT "                1 or 2 numbers only!GG"
1220 GO TO 1180
1230 J2=VAL(X$)
1240 IF J2>J1 AND J2<=16 THEN 1290
1250 PRINT "                    Ending Row must be greater than Beg Row"
1260 PRINT "                     and less then Row 16!GG"
1270 GO TO 1030

1280 REM ****************************** CHECK IF ROWS ARE ALREADY SPANNED

1290 J3=0
1300 FOR I1=J1 TO J2
1310     IF A2(I1)<>0 THEN 1330
1320     IF A3(I1)=0 THEN 1350
1330     J3=1
1340     I1=J2
1350 NEXT I1
1360 IF J3=0 THEN 1400
1370 PRINT "                    a Vert Desc already spans this Row!GG"
1380 GO TO 960

1390 REM ******** STORE SPANNED ROWS IN SINGLE & DOUBLE ARRAYS, A2 & A3
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1400 FOR I1=J1 TO J2
1410     IF W$="D" THEN 1440
1420     A2(I1)=1
1430     GO TO 1450
1440     A3(I1)=1
1450 NEXT I1
1460 A$=REP(Y$,A1(L4),LEN(Y$))
1470 A$=REP(X$,A1(L4+1),LEN(X$))
1480 A$=REP(W$,A1(L4+2),1)

1490 REM ********** L3 - NO OF SPACES AVAILABLE FOR VERTICAL DESCRIPTION

1500 L3=(J2-J1+1)*2+(J2-J1)
1510 IF W$="D" THEN 1540
1520 PRINT "      DESCRIPTION: ";
1530 GO TO 1550
1540 PRINT "   DESCRIPTION #1: ";
1550 GOSUB 2810
1560 GOSUB 2860
1570 INPUT X$
1580 IF LEN(X$)<=L3 AND LEN(X$)<>0 THEN 1610
1590 PRINT "                    ";L3;" characters maximum!GG"
1600 GO TO 1540
1610 A$=REP(X$,A1(L4+3),LEN(X$))
1620 IF W$="S" THEN 920
1630 PRINT "   DESCRIPTION #2: ";
1640 GOSUB 2810
1650 GOSUB 2860
1660 INPUT X$
1670 IF LEN(X$)<=L3 AND LEN(X$)<>0 THEN 1700
1680 PRINT "                    ";L3;" characters only!GG"
1690 GO TO 1630
1700 A$=REP(X$,A1(L4+4),LEN(X$))
1710 GO TO 920
1720 FOR I1=1 TO 300
1730 NEXT I1
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
 740 L4=28
1750 L1=0
1760 PRINT "LHORIZONTAL DESCRIPTIONSJ"
1770 L4=L4+2
1780 L1=L1+1
1790 IF L1>16 THEN 2042
1800 PRINT "J    DESCRIPTION: "
1810 PRINT "                  type '0' when complete"
1820 IF A2(L1)=0 THEN 1850
1830 L3=R4-4
1840 GO TO 1890
1850 IF A3(L1)=0 THEN 1880
1860 L3=R4-6
1870 GO TO 1890
1880 L3=R4-1
1890 PRINT "                  KK";
1900 GOSUB 2810
1910 GOSUB 2860
1920 INPUT X$
1930 IF X$="0" THEN 2042
1940 IF LEN(X$)<=L3 AND LEN(X$)<>0 THEN 1970
1950 PRINT "J                  ";INT(L3);" characters maximum!GG"
1960 GO TO 1800
1970 PRINT "J           LINE? _H";
1980 INPUT Y$
1990 IF Y$="Y" OR Y$="N" THEN 2020
2000 PRINT "                type a 'Y' or a 'N'!GG"
2010 GO TO 1970
2020 A$=REP(Y$,A1(L4),1)
2030 A$=REP(X$,A1(L4+1),LEN(X$))
2040 GO TO 1770

2041 REM ********************************** CHECK THAT # OF ROWS IS OKAY

2042 IF L1=>16 THEN 2056
2043 J1=0
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2044 FOR I1=L1 TO 16
2045      IF A2(I1)<>0 THEN 2047
2046      IF A3(I1)=0 THEN 2049
2047      J1=1
2048      I1=16
2049 NEXT I1
2050 IF J1=0 THEN 2056
2051 PRINT "JVertical Descriptions can not exceed Horizontal ";
2052 PRINT "Descriptions.  Start over!GG"
2053 GO TO 170

2055 REM ****************************************** STORE NO OF ROWS (L1)

2056 L1=L1-1
2057 X$=STR(L1)
2058 X$=REP("",1,1)
2059 A$=REP(X$,A1(9),LEN(X$))
2061 L1=L1+1
2062 FOR I1=1 TO 500
2063 NEXT I1

2068 REM ************************************* PREPARE FOR TAPE STORAGE

2069 PRINT "L"
2080 PRINT "JJWhat shall be the name of this Chart? __________";
2090 PRINT "HHHHHHHHHH";
2100 INPUT X$
2110 IF LEN(X$)<=10 AND LEN(X$)<>0 THEN 2150
2120 PRINT "Chart name can be a maximum of 10 characters!GG"
2130 GO TO 2080
2140 INPUT X$
2150 Z$=SEG(X$,1,1)
2160 IF Z$=>"A" AND Z$<="Z" THEN 2190
2170 PRINT "Chart name must begin with an Alphabetic character!GG"
2180 GO TO 2080
2190 FOR I1=2 TO LEN(X$)
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
2200     Z$=SEG(X$,I1,1)
2210     IF Z$=>"A" AND Z$<="Z" OR (Z$=>"0" AND Z$<="9") THEN 2240
2220     PRINT "No Blanks or Special Characters allowed for Chart name!GG"
2230     GO TO 2080
2240     P$=X$
2250     FIND 5
2270     J1=0
2280     J2=0
2290     FOR I1=1 TO 100
2300         READ @33:E$,E1
2322         IF E$<>"DONE" THEN 2330
2324         J1=3
2325         J2=I1
2326         I1=100
2328         GO TO 2346
2330         IF E$<>P$ THEN 2346
2332         J1=1
2334         J2=I1
2336         I1=100
2338         GO TO 2346
2346     NEXT I1
2350     IF J1=3 THEN 2420
2352     IF J1=1 THEN 2358
2356     GO TO 2420
2358     PRINT "J";P$;" already exists as a Chart name.  Should it be ";
2360     PRINT "over-written? _H";
2362     INPUT Z$
2364     IF Z$="N" THEN 2080
2366     IF Z$<>"Y" THEN 2358
2368     FIND E1
2370     READ @33:Z$
2372     Z$=SEG(Z$,A1(9),2)
2374     L2=VAL(Z$)
2376     IF L1-1=L2 THEN 2388
2378     PRINT "JTo properly access the associated Data File, this ";
2380     PRINT "Chart must have "
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
2381     PRINT L2;" Rows.  Start over!GG"
2382     FOR I1=1 TO 500
2384     NEXT I1
2386     GO TO 170
2388     FIND E1
2390     WRITE @33:A$
2392     CLOSE
2394     PRINT "JA Data File already exists for Chart ";P$;".  Should ";
2396     PRINT "it be "
2397     PRINT "initialized? _H";
2398     INPUT Z$
2400     IF Z$="N" THEN 2414
2402     IF Z$<>"Y" THEN 2394
2404     FIND E1+1
2406     DELETE D9
2408     DIM D9(L1-1,42)
2410     D9=-1
2412     WRITE @33:"01/01/50",L1-1,D9
2414     CLOSE
2416     PRINT "JJJJ";P$;" Chart format stored!GG"
2418     RETURN
2420     FIND 5
2421     IF J2=0 OR J2=1 THEN 2428
2422     FOR I1=1 TO J2-1
2424         READ @33:E$,E1
2426     NEXT I1
2427     E1=E1+2
2428     WRITE @33:P$,E1
2429     WRITE @33:"DONE",E1+2
2430     FIND E1
2432     WRITE @33:A$
2434     CLOSE
2436     FIND E1+1
2438     GO TO 2406
2440     FIND 5
2442     WRITE @33:P$,E1
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
2443     WRITE @33:"DONE",E1+1
2444     FIND 6
2446     WRITE @33:A$
2448     CLOSE
2450     FIND 7
2452     GO TO 2406

2710     REM *************** SUBROUTINE - ALL MEMBERS OF X$ MUST BE NUMERIC

2720     J1=0
2730     FOR K1=1 TO LEN(X$)
2740         Z$=SEG(X$,K1,1)
2750         IF Z$=>"0" AND Z$<="9" THEN 2780
2760         J1=1
2770         K1=LEN(X$)
2780     NEXT K1
2790     RETURN

2800     REM ******************** SUBROUTINE - PRINT L3 UNDERLINE CHARACTERS

2810     FOR K1=1 TO L3
2820         PRINT "_";
2830     NEXT K1
2840     RETURN

2850     REM ******************** SUBROUTINE - PRINT L3 BACKSPACE CHARACTERS

2860     FOR K1=1 TO L3
2870         PRINT "H";
2880     NEXT K1
2890     RETURN
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

TAPE VERSION -- FILE 3

```
100 INIT
110 PRINT "LThis Task allows Data Editing."
120 DIM H5(1,28),C1(4),C2(3),C3(3),C9(3),F$(51),G$(54),R$(24)
130 DIM W$(48),B2(3),B9(3),C$(9),C0(12),D2(3),V$(15),M$(1)
140 M$=CHR(13)

150 REM ***************** INITIALIZATION OF CALENDAR FUNCTION VARIABLES

160 DATA 31,28,31,30,31,30,31,31,30,31,30,31,1,7,1,3
170 RESTORE 160
180 READ C0,C1
190 F$="   JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC "
200 N3=-2.713E+288
210 V$=STR(N3)
220 W$="          SUN    MON   TUESWEDNES THURS    FRI SATUR"
230 G$="           DAYS       WEEKD      MONTHS     QUARTERS YEARS"

240 REM ********* READ FROM TAPE: R$ - DATE, L1 - # OF ROWS, D9 - DATA

250 PRINT "JJWhich Chart? _____HHHHHHHHHH";
260 INPUT X$
270 IF LEN(X$)<=10 AND LEN(X$)<>0 THEN 300
280 PRINT "Chart name must be less than 10 characters!GG"
290 GO TO 250
300 Z$=SEG(X$,1,1)
310 IF Z$=>"A" AND Z$<="Z" THEN 340
320 PRINT "Chart name must begin with an Alphabetic character!GG"
330 GO TO 250
340 FOR I1=2 TO LEN(X$)
350     Z$=SEG(X$,I1,1)
360     IF Z$=>"A" AND Z$<="Z" OR (Z$=>"0" AND Z$<="9") THEN 390
370     PRINT "No Blanks or Special characters allowed for Chart name!GG"
380     GO TO 250
390     O$=X$
400     FIND 5
408     J1=0
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
 12     FOR I1=1 TO 100
416         READ @33:E$,E1
424         IF E$<>"DONE" THEN 440
428         J1=2
432         I1=100
436         GO TO 468
440         IF X$<>E$ THEN 468
444         J1=1
448         I1=100
452         GO TO 468
468     NEXT I1
476     IF J1=1 THEN 492
480     IF J1=2 THEN 484
484     PRINT "JChart ";X$;" is not stored on this Tape!GG"
488     GO TO 250
492     FIND E1+1
496     READ @33:R$,L1
500     DELETE D9
504     DIM D9(L1,28)
508     FIND E1+1
512     READ @33:R$,L1,D9

570     REM ********************** P$ & R$ - DATE FOR FIRST COLUMN OF DATA

580     Z$=R$
590     PRINT "JJJBeginning Data Date is ";R$;".  Is this correct? ";
600     INPUT Y$
610     GOSUB 3380
620     P$=R$
630     IF Y$="Y" THEN 770
640     IF Y$<>"N" THEN 590
650     PRINT "JNew Date: _______ HHHHHHHH";
660     INPUT R$
670     GOSUB 3380
680     IF H1<>0 THEN 650
690     GOSUB 3040
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
,00     B$=SEG(W$,6*C7+1,6)
710     IF B$="   SUN" THEN 740
720     PRINT R$;" is a ";B$;".  Indicate a Sunday!GG"
730     GO TO 650
740     P$=R$
750     GOSUB 2390

760     REM ************************************************** DATA EDITING

770     PRINT "LDuring Data modification, type:J"
780     PRINT "  <CR> if present value is correct;"
790     PRINT "  -1 if value is not to print; or "
800     PRINT "  0 if it is to be calculated."
810     PRINT "JJJTyping a '0' for any of the following Questions will ";
820     PRINT "cause               an exit from the indicated Process."
830     R$=P$
840     PRINT "JJDo you want to go thru each Week(1), only selected ";
850     PRINT "Weeks(2),            or Neither(0)? _H";
860     INPUT Y$
870     IF Y$="0" THEN 1820
880     IF Y$="1" THEN 1130
890     IF Y$<>"2" THEN 840
900     R$=P$
910     GOSUB 3380
920     B2=C2
930     C3=C2
940     GOSUB 3380
950     GOSUB 3040
960     Z1=C8
970     PRINT "JWhich Date: _________HHHHHHHH";
980     INPUT R$
990     IF R$="0" THEN 770
1000     GOSUB 3380
1010     IF H1<>0 THEN 970
1020     B2=C2
1030     C3=C2
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
1040      GOSUB 3040
1050      Z1=C8-Z1
1060      IF Z1=>0 AND Z1<=27 THEN 1090
1070      PRINT R$;" is out of the 28-Week range!GG"
1080      GO TO 900
1090      I1=Z1+1
1100      GOSUB 1290
1110      GO TO 900

1120      REM ****************************************** LOOP THRU EACH COLUMN

1130      C5=0
1140      R$=P$
1150      GOSUB 3380
1160      FOR I1=1 TO 28

1170          REM ************************************* FIND DATE FOR THIS COLUMN

1180          GOSUB 2720
1190          PRINT M$;"JJ"
1200          PRINT "Should Data for ";C2(1);"/";C2(2);"/";C2(3);" be Modified?";
1210          PRINT " H";
1220          INPUT Y$
1230          IF Y$="0" THEN 1760
1240          IF Y$="N" THEN 1740
1250          IF Y$<>"Y" THEN 1200
1260          GOSUB 1290
1270          GO TO 1740

1280          REM ********************************************** LOOP THRU EACH ROW

1290          FOR I2=1 TO L1
1300              Y$=STR(D9(I2,I1))
1310              Y$=REP("",1,1)
1320              J1=POS(Y$,".",1)
1330              IF J1=0 THEN 1390
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
1340          Y$=REP("/",J1,1)
1350          Y$=REP("",LEN(Y$)-1,2)
1360          IMAGE 6A,2D,2A,6A,15A
1370          IF Y$<>"1/0E" THEN 1390
1380          Y$="0/00"
1390          PRINT USING 1360:"  Row ";I2;": ";Y$;"      HHHHHH"
1400          PRINT "                    K";
1410          INPUT X$
1415          IF X$="-1" THEN 1700
1420          IF LEN(X$)<=6 THEN 1450
1430          PRINT "                    6 Characters maximum!GG"
1440          GO TO 1390
1450          IF LEN(X$)=0 THEN 1720

1460          REM *** SUBROUTINE - ALL CHARACTERS OF X$ MUST BE NUMERIC OR A "/"

1470          J1=0
1480          FOR K1=1 TO LEN(X$)
1490              Z$=SEG(X$,K1,1)
1500              IF Z$=>"0" AND Z$<="9" OR Z$="/" THEN 1530
1510              J1=1
1520              K1=LEN(X$)
1530          NEXT K1
1540          IF J1=0 THEN 1580
1550          PRINT "                    Type only numbers and, optionally, ";
1560          PRINT " a '/'!GG"
1570          GO TO 1390
1580          J1=POS(X$,"/",1)
1590          IF J1=0 THEN 1700
1600          IF J1+3>LEN(X$) THEN 1650
1610          Z$=SEG(X$,J1+3,1)
1620          IF Z$=" " THEN 1650
1630          PRINT "Only two numbers allowed after the '/'!GG"
1640          GO TO 1390
1650          X$=REP(".",J1,1)
1660          U1=VAL(X$)
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
1670            U2=1.0E-4
1680            U1=U1+U2
1690            GO TO 1710
1700            U1=VAL(X$)
1710            D9(I2,I1)=U1
1720        NEXT I2
1730        RETURN
1740        C5=1
1750        GO TO 1770
1760        I1=42
1770    NEXT I1
1780    GO TO 770

1790    REM ********************************************** TOTAL CALCULATIONS
1800    REM *** L2 - NO OF ROWS TO BE ADDED TOGETHER * L3 - # OF TOTAL ROW
1810    REM * H2 - ARRAY OF ROW #s TO BE ADDED * H5 - ARRAY OF SUMMED DATA

1820    PRINT "LTotal Calculations"
1830    PRINT "JJHow many Rows? __ (type '0' when complete)"
1840    PRINT "                K";
1850    INPUT L2
1860    IF L2<=0 THEN 2260
1870    IF L2=>1 AND L2<=L1 THEN 1900
1880    PRINT "Only ";L1;" Rows exist!GG"
1890    GO TO 1830
1900    DELETE H2
1910    DIM H2(L2)
1920    PRINT "JIndicate Rows: ____________________________________________"
1930    PRINT "                K";
1940    INPUT H2
1950    PRINT "JIndicate Total Row #:  HH";
1960    INPUT L3
1970    IF L3=>1 AND L3<=L1 THEN 1990
1980    GO TO 1880
1990    J1=0
2000    FOR I1=1 TO L2
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
2010        IF H2(I1)=>1 AND H2(I1)<=L1 THEN 2060
2020        PRINT "Row #";H2(I1);" is outside Row Range.  Row #'s must ";
2030        PRINT "fall between 1 and ";L1;"!GG"
2040        J1=1
2050        GO TO 2100
2060        IF H2(I1)<>L3 THEN 2100
2070        PRINT "Row #";H2(I1);" to be Added is the same as the Total Row ";
2080        PRINT "#";L3;".                    They must be different!GG"
2090        J1=1
2100    NEXT I1
2110    IF J1=1 THEN 1830
2120    H5=0

2130    REM *************** ADD 1 ROW AT A TIME ACROSS ALL 42 DATA COLUMNS

2140    FOR I1=1 TO L2
2150        FOR I2=1 TO 28
2160            IF D9(H2(I1),I2)<0 THEN 2190
2170            H5(1,I2)=H5(1,I2)+D9(H2(I1),I2)
2180            GO TO 2200
2190            H5(1,I2)=-1
2200        NEXT I2
2210    NEXT I1
2220    FOR I1=1 TO 28
2230        D9(L3,I1)=H5(1,I1)
2240    NEXT I1
2250    GO TO 1820
2260    GOSUB 2390
2270    PRINT "LWould you like to go thru the Editing process again? _H";
2280    INPUT Y$
2290    IF Y$="Y" THEN 580
2300    IF Y$<>"N" THEN 2270
2310    FIND E1+1
2320    WRITE @33:P$,L1,D9
2330    CLOSE
2340    PRINT "JJBeginning Date, Number of Rows, and the Weekly Data ";
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
2350     PRINT "has been            stored in the Data File for Chart ";
2360     PRINT O$;"!GG"
2370     RETURN

2380     REM ************************************** SUBROUTINE - LIST DATA

2390     PRINT "LWould you like a Print-out of the existing Data? _H";
2400     INPUT Y$
2410     IF Y$="N" THEN 2660
2420     IF Y$<>"Y" THEN 2390
2430     R$=P$

2440     REM ************************************** LOOP THRU EACH COLUMN

2450     C5=0
2460     FOR I1=1 TO 28

2470         REM ********************************** FIND DATE FOR THIS COLUMN

2480         GOSUB 3380
2490         GOSUB 2720
2500         PRINT M$;"JJCOLUMN: ";I1;"  (";C2(1);"/";C2(2);"/";C2(3);")"

2510         REM ****************************************** LOOP THRU EACH ROW

2520         FOR I2=1 TO L1
2530             Y$=STR(D9(I2,I1))
2540             IF Y$<>" 1.0E-4" THEN 2570
2550             Y$="0/0"
2560             GO TO 2620
2570             Y$=REP("",1,1)
2580             I3=POS(Y$,".",1)
2590             IF I3=0 THEN 2620
2600             Y$=REP("/",I3,1)
2610             Y$=REP("",LEN(Y$)-1,2)
2620             PRINT "  R";I2;": ";Y$;"  ";
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2630          NEXT I2
2640          C5=C5+1
2650      NEXT I1
2660      PRINT M$;"JJJTo continue, type <CR>. ";
2670      INPUT Y$
2680      RETURN

2690      REM ******************** SUBROUTINE - DATE + OR - C5 TIME SEGMENTS
2700      REM ************************ C2=(mm,dd,yy)  C5=# of time segments
2710      REM ********************************** T1=time mesh * C6,C7=scratch

2720      IF C5=0 THEN 2820
2730      C6=SGN(C5)
2740      T1=2
2750      GO TO T1 OF 2760,2760,2950,2950,2970
2760      C2(2)=C2(2)+C1(T1)*C5
2770      C0(2)=28
2780      IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 2800
2790      C0(2)=29
2800      IF C2(2)<1 OR C2(2)>C0(C2(1)) THEN 2830
2810      C0(2)=28
2820      RETURN
2830      IF C6+1 THEN 2920
2840      C2(1)=C2(1)+C6
2850      IF C2(1)=>1 AND C2(1)<=12 THEN 2900
2860      C2(3)=C2(3)+C6
2870      C2(1)=C2(1)-12*C6
2880      IF T1>2 THEN 2850
2890      GO TO 2910
2900      IF T1>2 THEN 2990
2910      IF C6+1 THEN 2770
2920      C2(2)=C2(2)-C6*C0(C2(1))
2930      IF C6+1 THEN 2840
2940      GO TO 2770
2950      C2(1)=C2(1)+C1(T1)*C5
2960      GO TO 2850
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2970     C2(3)=C2(3)+C5
2980     GO TO 2810
2990     C2(2)=C2(2) MIN C0(C2(1))
3000     GO TO 2810

3010     REM *************** SUBROUTINE - DAY #, WEEK #, & DAY OF THE WEEK
3020     REM ********************** C4=day # * C8=week # * C7=day of week
3030     REM ********************************** C2=(mm,dd,yy) * B3=scratch

3040     C4=0.25*C2(3)=INT(0.25*C2(3)) AND C2(1)>2
3050     IF C2(1)=1 THEN 3090
3060     FOR B3=1 TO C2(1)-1
3070         C4=C4+C0(B3)
3080     NEXT B3
3090     C4=C4+C2(2)+365*C2(3)+INT(0.25*(C2(3)-1))
3100     C8=INT(C4/7)+1
3110     C7=C4-(C8-1)*7+1
3120     RETURN

3130     REM ********************* SUBROUTINE - TIME SEGMENTS BETWEEN DATES
3140     REM ************************************************** T1=time mesh
3150     REM ************** 1st date=B2=(mm,dd,yy) * c8=# of time segments
3160     REM **** 2nd date=C3=(mm,dd,yy) * B2,B3,C0,C2,C4,C5,C6,C7=scratch

3170     T1=2
3180     GO TO T1 OF 3190,3190,3300,3300,3330
3190     C2=C3
3200     GOSUB 3040
3210     C5=C4
3220     C6=C8
3230     C2=B2
3240     GOSUB 3040
3250     IF T1=2 THEN 3280
3260     C8=C5-C4
3270     RETURN
3280     C8=C6-C8
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
3290      RETURN
3300      C8=INT((C3(1)+2)/C1(T1))-INT((B2(1)+2)/C1(T1))
3310      C8=12/C1(T1)*(C3(3)-B2(3))+C8
3320      RETURN
3330      C8=C3(3)-B2(3)
3340      RETURN

3350      REM ******************************** SUBROUTINE - DATE ENTRY VERIFY
3360      REM ********************** H1=flag (118=bad 0=OK) * R$=date string
3370      REM ******* C2=(mm,dd,yy) * C7=pos of 1st "/" * C8=pos of 2nd "/"

3380      H1=0
3390      T1=2
3400      C7=POS(R$,"/",1)
3410      C8=POS(R$,"/",C7+1)
3420      C2(2)=1
3430      B$=SEG(R$,(C8 MAX C7)+1,LEN(R$)-(C8 MAX C7))
3440      B$=B$&V$
3450      C2(3)=VAL(B$)
3460      IF C2(3)<>N3 AND C2(3)=>1 AND C2(3)=INT(C2(3)) THEN 3490
3470      H1=118
3480      GO TO 3840
3490      IF C7<>0 THEN 3530
3500      IF T1<>5 THEN 3470
3510      C2(1)=1
3520      RETURN
3530      B$=SEG(R$,1,C7-1)
3540      IF B$<>"END" THEN 3590
3550      IF T1<>5 AND C8<>0 THEN 3470
3560      C2(1)=12
3570      C2(2)=31
3580      RETURN
3590      B$=B$&V$
3600      C2(1)=VAL(B$)
3610      IF C2(1)=N3 OR C2(1)>12 OR C2(1)<>INT(C2(1)) OR C2(1)<1 THEN 3470
3620      IF C8<>0 THEN 3680
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
3630     IF T1<3 OR T1=5 THEN 3470
3640     C2(1)=C1(T1)*(C2(1)-1)+1
3650     C2(1)=C2(1)+C9(1)-C1(T1)*INT(C9(1)/C1(T1)-0.2)-1
3660     IF C2(1)>12 THEN 3470
3670     RETURN
3680     C0(2)=28
3690     IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 3710
3700     C0(2)=29
3710     B$=SEG(R$,C7+1,C8-C7-1)
3720     IF B$<>"END" THEN 3780
3730     IF T1<3 OR T1>4 THEN 3470
3740     C2(1)=C1(T1)*C2(1)
3750     IF C2(1)>12 THEN 3470
3760     C2(2)=C0(C2(1))
3770     RETURN
3780     B$=B$&V$
3790     C2(2)=VAL(B$)
3800     IF C2(2)=N3 OR C2(2)<>INT(C2(2)) OR C2(2)<1 THEN 3470
3810     IF C2(2)>C0(C2(1)) THEN 3470
3820     C0(2)=28
3830     RETURN
3840     PRINT "Invalid Date!GG"
3850     RETURN
3860     FOR R1=1 TO 42
3870         FOR R2=1 TO L1
3880             PRINT "ROW ";R2;"COLUMN ";R1;": ";
3890             INPUT D9(R2,R1)
3900         NEXT R2
3910     NEXT R1
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

TAPE VERSION -- FILE 4

```
 00 INIT
110 PRINT "LThis Task generates GANTT Chart."
560 M9=MEMORY
570 DIM A1(61),A2(16),A3(16),A4(12),A5(12),K$(300),S(321)
580 DIM C1(4),C2(3),C3(3),C9(3),F$(51),G$(54),R$(24),V$(15),W$(48)
590 DIM B2(3),B9(3),C$(9),C0(12),D2(3),D5(3),H2(3)
600 DATA 31,28,31,30,31,30,31,31,30,31,30,31,1,7,1,3
610 N6=0
620 RESTORE 600
630 READ C0,C1
650 F$="   JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC "
660 N3=-2.713E+288
670 V$=STR(N3)
680 W$="          SUN   MON  TUESWEDNES THURS   FRI SATUR"
690 G$="           DAYS      WEEKS     MONTHS    QUARTERS YEARS"

710 REM .....strokes.....

720 DATA 0,0,0,7,4,7,5,5.25,4,3.5,5,1.75,4,0,0,0,2.5,7,5,0,4,3,1,3
730 DATA 2,0,2,1,3,1,3,0,2,0,5,0,0,0,0,3.5,2,3.5,0,3.5,0,7,5,7,4,7,4,2
740 DATA 3,0,1,0,0,1,5,7,0,7,0,0,5,0,5,3.5,3,3.5,0,0,0,7,3,7,5,5,5,2,3
750 DATA 0,0,0,7,0,3.5,5,3.5,5,7,5,0,0,0,0,7,5,7,5,0,3,2
760 DATA 2.5,0,2.5,7,0,7,5,7,0,0,5,0,5,3.5,0,3.5,0,7,5,7,5,4,0,0,5,0
770 DATA 0,5,7,2.5,3.5,0,7,5,0,0,0,7,2.5,0,5,7,5,0,5,7,0,7,0,3.5,5,3.5
780 DATA 5,7,0,7,0,0,5,0,5,3.5,0,3.5,0,0,0,7,5,7,5,3.5,0,3.5,3,3.5,5,0
790 DATA 0,0,7,5,0,5,7,1,0,2.5,0,2.5,7,1,7,4,7,2.5,7,2.5,0,4,0
800 DATA 0,0,3.5,5,7,0,3.5,5,0,0,3.5,0,7,0,0,5,0,5,7,0,7,0,0,2.5,3.5,5
810 DATA 0,5,7,3,0,3,7,0,2,5,2

820 REM .....begins.....

830 DATA 25,0,1,3,15,35,37,45,59,61,59,71,82,92,92,92,104,108,108
840 DATA 112,112,120,129,129,138,140,146,158,154,168,168,181,189,191
850 DATA 189,197,204,216,224,234

860 REM .....ends.....
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
870 DATA 34,0,16,16,24,48,48,58,66,66,70,84,93,101,101,103,111,115,113
880 DATA 123,123,129,138,136,147,145,155,167,167,177,181,188,196,194
890 DATA 204,204,217,223,233,241
900 RESTORE 720
910 READ S
930 G1=0
940 G2=0
950 G3=130
960 G4=100
990 G5=5
1000 G6=3
1010 G7=3
1020 G8=0.5
1050 R1=3
1060 R2=5
1070 R3=3
1080 R4=31.5
1130 E1=1.7
1140 E2=2.5
1150 E3=1.825
1160 E4=3.5
1170 E5=1
1180 E6=2.5
1190 E7=1
1200 E8=2.5
1210 E9=0.866
1220 E0=1.5
1240 DATA 1,16,31,76,92,98,128,158,166,168,170,172,173,203,233,235
1250 DATA 237,238,268,298,300,302,303,333,363,365,367,368,398,428,429
1260 DATA 459,460,490,491,521,522,552,553,583,584,614,615,645,646,676
1270 DATA 677,707,708,738,739,769,770,800,801,831,832,862,863,893,894
1280 RESTORE 1240
1290 READ A1
1310 FOR I1=1 TO 16
1320     A2(I1)=0
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1330      A3(I1)=0
1340 NEXT I1
1370 PRINT "JJWhat is the Name of the GANTT Chart? __________"
1380 PRINT "                                        K";
1390 INPUT P$
1400 FIND 5
1404 J1=0
1406 FOR I1=1 TO 100
1408      READ @33:E$,E1
1412      IF E$<>"DONE" THEN 1420
1414      J1=2
1416      I1=100
1418      GO TO 1434
1420      IF P$<>E$ THEN 1434
1422      J1=1
1424      I1=100
1426      GO TO 1434
1434 NEXT I1
1438 IF J1=1 THEN 1446
1440 IF J1=2 THEN 1442
1442 PRINT "J";P$;" Chart does not exist on this Tape!GG"
1444 GO TO 1370
1446 FIND E1
1510 DELETE 1,1500
1512 DIM A$(923)
1514 READ @33:A$
1516 FIND E1+1
1518 READ @33:R$,L1
1600 DELETE 1400,1590
1602 M9=MEMORY
1604 DELETE D9
1606 DIM D9(L1,28)
1608 FIND E1+1
1610 READ @33:R$,L1,D9
1670 GOSUB 9050
1680 GOSUB 8720
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
.690 Z1=C8
1700 Z$=R$
1710 PRINT "JJBeginning Data Date is ";Z$
1720 PRINT "Indicate beginning Column Date: _______HHHHHHHH";
1730 INPUT R$
1740 GOSUB 9050
1750 IF H1 THEN 1720
1760 GOSUB 8720
1770 Z1=C8-Z1
1780 IF Z1=>0 AND Z1<=27 THEN 1830
1790 PRINT "Beginning Date must be within 28 weeks of ";Z$
1800 PRINT R$;" is ";Z1;" weeks from ";Z$
1810 R$=Z$
1820 GO TO 1670
1830 B$=SEG(W$,6*C7+1,6)
1840 IF B$="   SUN" THEN 1880
1850 PRINT R$;" is ";B$;". Select a Sunday!GG"
1860 R$=Z$
1870 GO TO 1670
1880 N5=Z1
1890 I2=0
1910 PRINT "LRow Descriptions for Chart ";P$;"J"
1920 FOR I1=30 TO 61 STEP 2
1930     I2=I2+1
1940     Y$=SEG(A$,A1(I1),1)
1950     IF Y$=" " THEN 2000
1960     X$=SEG(A$,A1(I1+1),30)
1970     PRINT "  ROW ";I2;": ";X$
1980     GO TO 2000
2000 NEXT I1
2010 PRINT "JJIndicate the 3 Rows to be Graphed: _______HHHHHHHH"
2020 PRINT "(If necessary, type 0's to make 3) K";
2030 INPUT H2
2040 I2=0
2050 FOR I1=1 TO 3
2060     IF H2(I1)=0 THEN 2100
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
_070     IF H2(I1)=>1 AND H2(I1)<=L1+1 THEN 2100
2080     I1=3
2090     I2=1
2100 NEXT I1
2110 IF I2=0 THEN 2150
2120 PRINT "Rows must be between 1 and ";L1;"!GG"
2130 GO TO 2010
2150 PRINT "LLegend name for:J"
2160 IF H2(1)=0 THEN 2220
2170 PRINT "  Row ";H2(1);": _____HHHHHHHHHHHHHHH";
2180 INPUT O$
2190 IF LEN(O$)<=15 AND LEN(O$)<>0 THEN 2220
2200 PRINT "         15 characters maximum!GG"
2210 GO TO 2170
2220 IF H2(2)=0 THEN 2280
2230 PRINT "  Row ";H2(2);": _____HHHHHHHHHHHHHHH";
2240 INPUT P$
2250 IF LEN(P$)<=15 AND LEN(P$)<>0 THEN 2280
2260 PRINT "         15 characters maximum!GG"
2270 GO TO 2230
2280 IF H2(3)=0 THEN 2350
2290 PRINT "  Row ";H2(3);": _____HHHHHHHHHHHHHHH";
2300 INPUT Q$
2310 IF LEN(Q$)<=15 AND LEN(Q$)<>0 THEN 2350
2320 PRINT "         15 characters maximum!GG"
2330 GO TO 2290
2350 N1=0
2360 FIND E1+1
2365 DELETE 1510,2360
2370 DELETE D9
2375 M9=MEMORY
2380 DIM D9(L1,28)
2390 READ @33:Y$,I1,D9
2410 N1=0
2420 N2=9999999
2430 Z2=Z1+13 MIN 27
```

TITLE

Gantt Chart

ABSTRACT NO

51/07-0102/0

```
2440 FOR I2=1 TO 3
2450     FOR I1=Z1+1 TO Z2+1
2460         IF H2(I2)=0 THEN 2530
2470         N3=D9(H2(I2),I1)
2480         IF N3<=N1 THEN 2500
2490         N1=N3
2500         IF N3=>N2 THEN 2520
2510         N2=N3
2520         GO TO 2540
2530         I1=Z2+1
2540     NEXT I1
2550 NEXT I2
2560 IF N2=>0 THEN 2580
2570 N2=0
2580 PRINT "JJJThe minimum Y value of the selected Rows is ";INT(N2)
2590 PRINT "and the maximum Y value is ";INT(N1)
2600 PRINT "JIndicate graph's:"
2610 PRINT "  Minimum Y value: _____HHHHHH";
2620 INPUT H5
2630 PRINT "  Maximum Y value: _____HHHHHH";
2640 INPUT H6
2650 PRINT "  Interval       : _____HHHHHH";
2660 INPUT H7
2670 I3=(H6-H5)/H7
2680 I3=I3-INT(I3)
2690 IF I3=0 THEN 2720
2700 PRINT "Interval must evenly divide into the range!GG"
2710 GO TO 2580
2720 PRINT "JJJShall the Chart be generated on the Screen(32) or on"
2730 PRINT "the Plotter(1)? __HH";
2740 INPUT D1
2750 IF D1=32 OR D1=1 THEN 2780
2760 PRINT "                type '32' or '1' please!GG"
2770 GO TO 2720
2780 IF D1=32 THEN 2840
2790 PRINT "JReady Plotter!GG"
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2800 PRINT "To continue, type <CR>. ";
2810 INPUT Y$
2840 PRINT @32:"L"
2850 Z9=G5+G6*2+G7+G8
2860 MOVE @D1:G1,G4-Z9-L1*R1
2870 DRAW @D1:G3,G4-Z9-L1*R1
2880 DRAW @D1:G3,G4
2890 DRAW @D1:G1,G4
2900 DRAW @D1:G1,G4-Z9-L1*R1
2910 MOVE @D1:G1,G4-G5
2920 DRAW @D1:G3,G4-G5
2930 MOVE @D1:R4+G8,G4-G5-G6
2940 DRAW @D1:G3,G4-G5-G6
2950 MOVE @D1:R4+G8,G4-G5-G6*2
2960 DRAW @D1:G3,G4-G5-G6*2
2970 MOVE @D1:G1,G4-G5-G6*2-G7
2980 DRAW @D1:G3,G4-G5-G6*2-G7
2990 MOVE @D1:G1,G4-G5-G6*2-G7-G8
3000 DRAW @D1:G3,G4-G5-G6*2-G7-G8
3010 MOVE @D1:R4,G4-G5
3020 DRAW @D1:R4,G4-G5-G6*2-G7-G8-L1*R1
3030 MOVE @D1:R4+G8,G4-G5
3040 DRAW @D1:R4+G8,G4-G5-G6*2-G7-G8-L1*R1
3060 IF D1=32 THEN 3100
3070 PRINT @D1,25:90,90
3080 PRINT @D1,17:E5,E6
3090 GO TO 3120
3100 X6=E5
3110 Y6=E6
3120 FOR I1=10 TO 28 STEP 5
3130     V$=SEG(A$,A1(I1),2)
3140     IF V$="  " THEN 3410
3150     Z2=VAL(V$)
3160     U$=SEG(A$,A1(I1+1),2)
3170     Z3=VAL(U$)
3180     X$=SEG(A$,A1(I1+2),1)
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
3190      Y$=SEG(A$,A1(I1+3),30)
3210      IF X$="S" THEN 3340
3220      Z$=SEG(A$,A1(I1+4),30)
3240      MOVE @D1:5,G4-Z9-R1*(Z2-1)
3250      DRAW @D1:5,G4-Z9-R1*Z3
3270      IF D1=32 THEN 3430
3280      MOVE @D1:2,G4-Z9-R1*Z3+0.5
3290      PRINT @D1:Y$
3300      MOVE @D1:4.5,G4-Z9-R1*Z3+0.5
3310      PRINT @D1:Z$
3320      GO TO 3430
3340      MOVE @D1:3,G4-Z9-R1*(Z2-1)
3350      DRAW @D1:3,G4-Z9-R1*Z3
3370      IF D1=32 THEN 3430
3380      MOVE @D1:2.5,G4-Z9-R1*Z3+0.5
3390      PRINT @D1:Y$
3400      GO TO 3430
3410      I1=28
3420      GO TO 3490
3430      FOR I3=Z2 TO Z3
3440          IF X$="S" THEN 3470
3450          A3(I3)=1
3460          GO TO 3480
3470          A2(I3)=1
3480      NEXT I3
3490 NEXT I1
3500 IF D1=32 THEN 3530
3510 PRINT @D1,25:0,0
3530 GO TO 3760
3540 FOR I1=10 TO 28 STEP 6
3550      X$=SEG(A$,A1(I1),2)
3560      Y$=SEG(A$,A1(I1+1),2)
3570      Z$=SEG(A$,A1(I1+2),1)
3580      Z2=VAL(X$)
3590      Z3=VAL(Y$)
3610      IF Z$="  " THEN 3760
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
3630      IF Z$="2" THEN 3710
3660      FOR I2=Z2 TO Z3
3670          A2(I2)=1
3680      NEXT I2
3690      GO TO 3740
3710      FOR I2=Z2 TO Z3
3720          A3(I2)=2
3730      NEXT I2
3740 NEXT I1
3760 I2=0
3770 FOR I1=30 TO 61 STEP 2
3780      X$=SEG(A$,A1(I1),1)
3800      Y$=SEG(A$,A1(I1+1),30)
3810      I2=I2+1
3815      IF I2>L1 THEN 4020
3820      IF A2(I2)=0 THEN 3850
3830      Z1=G1+3.5
3840      GO TO 3890
3850      IF A3(I2)=0 THEN 3880
3860      Z1=G1+5.5
3870      GO TO 3890
3880      Z1=G1+0.5
3890      IF D1<>32 THEN 3950
3900      X7=Z1
3910      Y7=G4-Z9-R1*I2+0.5
3920      K$=Y$
3930      GOSUB 8240
3940      GO TO 3980
3950      PRINT @D1,17:E5,E6
3960      MOVE @D1:Z1,G4-Z9-R1*I2+0.5
3970      PRINT @D1:Y$
3980      IF X$="N" OR X$<>"Y" THEN 4030
3990      MOVE @D1:G1,G4-Z9-R1*I2
4000      DRAW @D1:G3,G4-Z9-R1*I2
4010      GO TO 4030
4020      I1=61
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-C102/0

```
4030 NEXT I1
4050 X$=SEG(A$,A1(1),15)
4060 Y$=SEG(A$,A1(2),15)
4070 IF D1=32 THEN 4100
4080 PRINT @D1,17:E1,E2
4090 GO TO 4120
4100 X6=E1
4110 Y6=E2
4120 IF Y$="                    " THEN 4380
4130 IF D1=32 THEN 4230
4140 MOVE @D1:0.5,G4-2.5
4150 PRINT @D1:X$
4160 MOVE @D1:0.6,G4-2.4
4170 PRINT @D1:X$
4180 MOVE @D1:0.5,G4-4.7
4190 PRINT @D1:Y$
4200 MOVE @D1:0.6,G4-4.6
4210 PRINT @D1:Y$
4220 GO TO 4510
4230 Y7=G4-2.4
4240 X7=0.5
4242 X6=E3
4244 Y6=E3
4250 K$=X$
4260 GOSUB 8240
4270 Y7=G4-2.3
4280 X7=0.6
4290 GOSUB 8240
4300 Y7=G4-4.6
4310 X7=0.5
4320 K$=Y$
4330 GOSUB 8240
4340 Y7=G4-4.5
4350 X7=0.6
4360 GOSUB 8240
4370 GO TO 4510
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
4380 IF D1=32 THEN 4450
4390 MOVE @D1:0.5,G4-4
4400 PRINT @D1,17:E3,E4
4410 PRINT @D1:X$
4420 MOVE @D1:0.6,G4-3.9
4430 PRINT @D1:X$
4440 GO TO 4510
4450 X7=0.5
4460 Y7=G4-4
4470 X6=E3
4480 Y6=E4
4490 K$=X$
4500 GOSUB 8240
4510 IF D1=32 THEN 4560
4520 PRINT @D1,17:E5,E6
4530 MOVE @D1:8,G4-13
4540 PRINT @D1:"A C T I V I T Y"
4550 GO TO 4620
4560 X7=8
4570 X6=E5
4580 Y6=E6
4590 Y7=G4-13
4600 K$="A C T I V I T Y"
4610 GOSUB 8240
4620 X$=SEG(A$,A1(4),16)
4630 IF D1=32 THEN 4670
4640 MOVE @D1:G3-17,G4-4
4650 PRINT @D1:X$
4660 GO TO 4710
4670 X7=G3-17
4680 Y7=G4-4
4690 K$=X$
4700 GOSUB 8240
4710 X$=SEG(A$,A1(3),45)
4720 GOSUB 7670
4730 IF D1=32 THEN 4780
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
 4740 PRINT @D1,17:E3,E4
4750 MOVE @D1:25.5,G4-4
4760 PRINT @D1:X$
4770 GO TO 4840
4780 X6=E3
4790 Y6=E4
4800 X7=25.5
4810 Y7=G4-4
4820 K$=X$
4830 GOSUB 8240
4840 X$=SEG(A$,A1(6),30)
4850 Y$=SEG(A$,A1(7),30)
4860 IF D1=32 THEN 4930
4870 PRINT @D1,17:E9,E0
4880 MOVE @D1:0.5,0
4890 PRINT @D1:Y$
4900 MOVE @D1:0.5,1.5
4910 PRINT @D1:X$
4920 GO TO 5030
4930 X6=E9
4940 Y6=E0
4950 X7=0.5
4960 Y7=0
4970 K$=Y$
4980 GOSUB 8240
4990 Y7=1.5
5000 X7=0.5
5010 K$=X$
5020 GOSUB 8240
5030 X$=SEG(A$,A1(5),6)
5040 IF D1=32 THEN 5080
5050 MOVE @D1:G3-5.7,2.5
5060 PRINT @D1:X$
5070 GO TO 5140
5080 X7=G3-5.7
5090 Y7=2
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
 ,00 K$=X$
5110 GOSUB 8240
5140 MOVE @D1:R4+G8,4
5150 DRAW @D1:R4+G8,G4-Z9-L1*R1-G8
5160 MOVE @D1:G3,G4-Z9-L1*R1-G8
5170 DRAW @D1:G3,4
5180 IF D1=32 THEN 5210
5190 PRINT @D1,17:E7,E8
5200 GO TO 5260
5210 X6=E7
5220 Y6=E8
5260 J4=0
5270 J5=0
5280 Z1=N5
5290 Z1=Z1+1
5300 Z2=Z2+1
5310 Z2=Z1+13 MIN 42
5320 Z3=(G3-R4-G8)/(Z2-Z1+1)
5330 Z5=R4+G8+0.5-Z3
5340 Z7=G4-Z9+0.5
5360 FOR I1=Z1 TO Z2
5370     J4=J4+1
5380     J5=J5+1
5390     Z8=Z7
5400     Z5=Z5+Z3
5420     FOR I2=1 TO L1
5430         Z8=Z8-R1
5450         IF D9(I2,I1)<0 THEN 5790
5460         Y$=STR(D9(I2,I1))
5470         IF Y$<>" 1.0E-4" THEN 5500
5480         Y$="  0/00"
5490         GO TO 5680
5500         Y$=REP("",1,1)
5510         I3=POS(Y$,".",1)
5520         IF I3=0 THEN 5580
5530         IF LEN(Y$)=>8 THEN 5640
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
5540          FOR I3=1 TO 8-LEN(Y$)
5550              Y$=" "&Y$
5560          NEXT I3
5570          GO TO 5640
5580          IF LEN(Y$)=>6 THEN 5640
5590          FOR I3=1 TO 6-LEN(Y$)
5600              Y$=" "&Y$
5610          NEXT I3
5620          GO TO 5680
5640          I3=POS(Y$,".",1)
5650          IF I3=0 THEN 5680
5660          Y$=REP("/",I3,1)
5670          Y$=REP("",LEN(Y$)-1,2)
5680          IF D1=32 THEN 5730
5690          PRINT @D1,17:E7,E8
5700          MOVE @D1:Z5,Z8
5710          PRINT @D1:Y$
5720          GO TO 5790
5730          X6=E7
5740          Y6=E8
5750          X7=Z5
5760          Y7=Z8
5770          K$=Y$
5780          GOSUB 8240
5790      NEXT I2
5810      IF D1<>32 THEN 5880
5820      X7=Z5+Z3-4.5
5830      Y7=G4-Z9+G8+0.5
5840      K$=STR(C2(2))
5850      K$=REP("",1,1)
5860      GOSUB 8240
5870      GO TO 5910
5880      MOVE @D1:Z5+Z3-4.5,G4-Z9+G8+0.5
5890      PRINT @D1:C2(2)
5910      D5=C2
5920      C5=1
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
5930      GOSUB 8400
5940      IF D5(1)=C2(1) THEN 6330
5950      IF D1<>32 THEN 6040
5960      X7=Z5-(J4*Z3/2-1.5)+Z3/2
5970      Y7=G4-Z9+G8+0.5+G7
5980      B$=SEG(F$,4*C2(1)-4,4)
5990      IF B$<>"   J" THEN 6010
6000      B$="DEC"
6010      K$=B$
6020      GOSUB 8240
6030      GO TO 6100
6040      MOVE @D1:Z5-(J4*Z3/2-1.5)+Z3/2,G4-Z9+G8+G7+0.5
6050      B$=SEG(F$,4*C2(1)-4,4)
6060      IF B$<>"   J" THEN 6080
6070      B$="DEC"
6080      PRINT @D1:B$
6090      IF Z5+Z3-0.7>128 THEN 6170
6100      J4=0
6120      MOVE @D1:Z5+Z3-0.5,G4-G5-G6
6130      DRAW @D1:Z5+Z3-0.5,G4-Z9-L1*R1
6140      MOVE @D1:Z5+Z3-0.5,G4-Z9-L1*R1-G8
6150      DRAW @D1:Z5+Z3-0.5,4
6170      IF D5(3)=C2(3) THEN 6330
6180      IF D1<>32 THEN 6260
6190      X7=Z5-J5*Z3/2-3+Z3
6200      Y7=G4-G5-G6+0.5
6210      K$=STR(D5(3))
6220      K$="19"&K$
6230      K$=REP("",3,1)
6240      GOSUB 8240
6250      GO TO 6290
6260      MOVE @D1:Z5-J5*Z3/2-3+Z3,G4-G5-G6+0.5
6270      PRINT @D1:"19";D5(3)
6280      IF Z5+Z3-0.5>128 THEN 6330
6290      J5=0
6310      MOVE @D1:Z5+Z3-0.5,G4-G5-G6
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
6320     DRAW @D1:Z5+Z3-0.5,G4-G5
6330 NEXT I1
6350 IF D5(1)<>C2(1) THEN 6510
6360 IF D1<>32 THEN 6450
6370 X7=Z5-(J4*Z3/2-1.5)+Z3/2
6380 Y7=G4-Z9+G8+G7+0.5
6390 B$=SEG(F$,4*C2(1),4)
6400 IF B$<>"   J" THEN 6490
6410 B$="DEC"
6420 K$=B$
6430 GOSUB 8240
6440 GO TO 6510
6450 MOVE @D1:Z5-(J4*Z3/2-1.5)+Z3/2,G4-Z9+G8+G7+0.5
6460 B$=SEG(F$,4*C2(1),4)
6470 IF B$<>"   J" THEN 6490
6480 B$="DEC"
6490 PRINT @D1:B$
6510 IF D5(3)<>C2(3) THEN 6650
6520 IF D1<>32 THEN 6600
6530 X7=Z5-J5*Z3/2-3+Z3
6540 Y7=G4-G5-G6+0.5
6550 K$=STR(D5(3))
6560 K$=REP("",1,1)
6570 K$="19"&K$
6580 GOSUB 8240
6590 GO TO 6650
6600 MOVE @D1:Z5-J5*Z3/2+Z3-3,G4-G5-G6+0.5
6610 PRINT @D1:"19";D5(3)
6650 Z5=R4+G8-0.5-Z3
6660 H3=4
6670 H4=G4-Z9-L1*R1-G8
6680 Z5=R4+G8
6690 H8=(H4-H3)/((H6-H5)/H7)
6700 H9=H5-H7
6730 FOR I1=H3 TO H4 STEP H8
6740     H9=H9+H7
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
  50      Y$=STR(H9)
6760      Y$=REP("",1,1)
6770      IF LEN(Y$)=6 THEN 6880
6780      FOR I3=1 TO 6-LEN(Y$)
6790          Y$=" "&Y$
6800      NEXT I3
6810      IF D1<>32 THEN 6870
6820      X7=R4+G8-6.6
6830      Y7=I1-1.25
6840      K$=Y$
6850      GOSUB 8240
6860      GO TO 6890
6870      MOVE @D1:R4+G8-6.6,I1-1.25
6880      PRINT @D1:Y$
6890      MOVE @D1:R4,I1
6900      DRAW @D1:G3,I1
6910 NEXT I1
6930 FOR I1=1 TO 3
6940      IF H2(I1)=0 THEN 7630
6970      IF Z1<>1 THEN 7020
6980      M1=-1
6990      M2=-1
7000      M3=Z5
7010      GO TO 7070
7020      M1=Z5
7030      M2=INT(D9(H2(I1),Z1-1))
7040      M2=H3+(M2-H5)/(H6-H5)*(H4-H3)
7050      M3=Z5
7070      FOR I2=Z1 TO Z2
7080          M3=M3+Z3
7090          M4=INT(D9(H2(I1),I2))
7100          M4=H3+(M4-H5)/(H6-H5)*(H4-H3)
7110          GOSUB 7840
7120          M1=M3
7130          M2=M4
7140      NEXT I2
```

TITLE

Gantt Chart

ABSTRACT NO:

51/07-0102/0

```
 160     M9=G4-L1*R1-G8-5-Z9
7170     IF I1=3 THEN 7490
7180     IF I1=2 THEN 7340
7190     M1=0.5
7200     M2=M9+1
7210     M3=5.5
7220     M4=M9+1
7230     N6=1
7240     GOSUB 7840
7250     IF D1=32 THEN 7290
7260     MOVE @D1:6,M9
7270     PRINT @D1:O$
7280     GO TO 7630
7290     X7=6
7300     Y7=M9
7310     K$=O$
7320     GOSUB 8240
7330     GO TO 7630
7340     M1=0.5
7350     M2=M9-3+1
7360     M3=5.5
7370     M4=M9-3+1
7380     N6=1
7390     GOSUB 7840
7400     IF D1=32 THEN 7440
7410     MOVE @D1:6,M9-3
7420     PRINT @D1:P$
7430     GO TO 7630
7440     X7=6
7450     Y7=M9-3
7460     K$=P$
7470     GOSUB 8240
7480     GO TO 7630
7490     M1=0.5
7500     M2=M9-6+1
7510     M3=5.5
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
7520      M4=M9-6+1
7530      N6=1
7540      GOSUB 7840
7550      IF D1=32 THEN 7590
7560      MOVE @D1:6,M9-6
7570      PRINT @D1:Q$
7580      GO TO 7630
7590      X7=6
7600      Y7=M9-6
7610      K$=Q$
7620      GOSUB 8240
7630 NEXT I1
7640 PRINT "GGGGG"
7645 INPUT Z$
7646 FIND 1
7647 OLD
7650 RETURN
7670 K2=LEN(X$)
7680 FOR K1=LEN(X$) TO 1 STEP -1
7690      Z$=SEG(X$,K1,1)
7700      IF Z$<>" " THEN 7730
7710      K2=K2-1
7720      GO TO 7740
7730      K1=1
7740 NEXT K1
7750 IF K2=LEN(X$) THEN 7820
7760 Z$=" "
7770 FOR K1=1 TO (LEN(X$)-K2)/2-1
7780      Z$=Z$&" "
7790 NEXT K1
7800 X$=REP(Z$,K2+1,(LEN(X$)-K2)/2)
7810 X$=REP(Z$,1,0)
7820 RETURN
7840 IF M1<0 OR M2<0 OR M3<0 OR M4<0 THEN 8220
7850 IF M2>H4 OR M4>H4 THEN 8220
7860 IF M2<H3 OR M4<H3 THEN 8220
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
7870 M2=INT(M2)
7880 M4=INT(M4)
7890 IF N6=1 THEN 7910
7900 IF D9(H2(I1),I2)<0 OR D9(H2(I1),I2-1)<0 THEN 8220
7910 MOVE @D1:M1,M2
7920 IF I1=1 THEN 8200
7930 S1=M3-M1
7940 S2=M4-M2
7950 S3=SQR(S1↑2+S2↑2)
7960 L=1
7970 IF ABS(S2)<L AND ABS(S1)<L THEN 8210
7980 U1=L*(S1/S3)
7990 U2=L*(S2/S3)
8000 U3=2*U1
8010 U4=2*U2
8020 GO TO I1 OF 8040,8090,8150
8030 GO TO 8200
8040 FOR I4=0 TO M3-M1-U1*2 MAX M4-M2-U2*2 STEP 2*(U1 MAX U2)
8050      RDRAW @D1:U1,U2
8060      RMOVE @D1:U1,U2
8070 NEXT I4
8080 GO TO 8200
8090 FOR I4=0 TO M3-M1-U1 MAX M4-M2-U2 STEP U1 MAX U2
8100      RDRAW @D1:0,0
8110      RMOVE @D1:U1,U2
8120 NEXT I4
8130 MOVE @D1:M3,M4
8140 GO TO 8210
8150 FOR I4=0 TO M3-M1-U1*3 MAX M4-M2-U2*3 STEP 3*(U1 MAX U2)
8160      RDRAW @D1:U3,U4
8170      RMOVE @D1:U1,U2
8180 NEXT I4
8190 DRAW @D1:M3,M4
8200 DRAW @D1:M3,M4
8210 N6=0
8220 RETURN
```

TITLE	ABSTRACT NO
Gantt Chart	51/07-0102/0

```
8240 FOR I=1 TO LEN(K$)
8250     J$=SEG(K$,I,1)
8260     J=POS(". B3AEFJCLGDHOOQTZ7S52XYMV968PRN]1I[KUW4",J$,1)
8270     IF J>0 THEN 8290
8280     J=1
8290     IF S(J+241)=0 THEN 8340
8300     MOVE X7+S(S(J+241))*X6*0.1333,Y7+S(S(J+241)+1)*Y6*0.087
8310     FOR J1=S(J+241)+2 TO S(J+281) STEP 2
8320         PRINT @32,20:S(J1)*X6*0.1333+X7,S(J1+1)*Y6*0.087+Y7
8330     NEXT J1
8340     X7=X7+X6
8350 NEXT I
8360 RETURN
8400 IF C5=0 THEN 8500
8410 C6=SGN(C5)
8420 T1=2
8430 GO TO T1 OF 8440,8440,8630,8630,8650
8440 C2(2)=C2(2)+C1(T1)*C5
8450 C0(2)=28
8460 IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 8480
8470 C0(2)=29
8480 IF C2(2)<1 OR C2(2)>C0(C2(1)) THEN 8510
8490 C0(2)=28
8500 RETURN
8510 IF C6+1 THEN 8600
8520 C2(1)=C2(1)+C6
8530 IF C2(1)=>1 AND C2(1)<=12 THEN 8580
8540 C2(3)=C2(3)+C6
8550 C2(1)=C2(1)-12*C6
8560 IF T1>2 THEN 8530
8570 GO TO 8590
8580 IF T1>2 THEN 8670
8590 IF C6+1 THEN 8450
8600 C2(2)=C2(2)-C6*C0(C2(1))
8610 IF C6+1 THEN 8520
8620 GO TO 8450
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
  30 C2(1)=C2(1)+C1(T1)*C5
8640 GO TO 8530
8650 C2(3)=C2(3)+C5
8660 GO TO 8490
8670 C2(2)=C2(2) MIN C0(C2(1))
8680 GO TO 8490
8720 C4=0.25*C2(3)=INT(0.25*C2(3)) AND C2(1)>2
8730 IF C2(1)=1 THEN 8770
8740 FOR B3=1 TO C2(1)-1
8750     C4=C4+C0(B3)
8760 NEXT B3
8770 C4=C4+C2(2)+365*C2(3)+INT(0.25*(C2(3)-1))
8780 C8=INT(C4/7)+1
8790 C7=C4-(C8-1)*7+1
8800 RETURN
8850 GO TO T1 OF 8860,8860,8970,8970,9000
8860 C2=C3
8870 GOSUB 3970
8880 C5=C4
8890 C6=C8
8900 C2=B2
8910 GOSUB 3970
8920 IF T1=2 THEN 8950
8930 C8=C5-C4
8940 RETURN
8950 C8=C6-C8
8960 RETURN
8970 C8=INT((C3(1)+2)/C1(T1))-INT((B2(1)+2)/C1(T1))
8980 C8=12/C1(T1)*(C3(3)-B2(3))+C8
8990 RETURN
9000 C8=C3(3)-B2(3)
9010 RETURN
9050 H1=0
9060 C7=POS(R$,"/",1)
9070 C8=POS(R$,"/",C7+1)
9080 C2(2)=1
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
9090 B$=SEG(R$,(C8 MAX C7)+1,LEN(R$)-(C8 MAX C7))
9100 B$=B$&V$
9110 C2(3)=VAL(B$)
9120 IF C2(3)<>N3 AND C2(3)=>1 AND C2(3)=INT(C2(3)) THEN 9150
9130 E1=118
9140 GO TO 9500
9150 IF C7<>0 THEN 9190
9160 IF T1<>5 THEN 9130
9170 C2(1)=1
9180 RETURN
9190 B$=SEG(R$,1,C7-1)
9200 IF B$<>"END" THEN 9250
9210 IF T1<>5 AND C8<>0 THEN 9130
9220 C2(1)=12
9230 C2(2)=31
9240 RETURN
9250 B$=B$&V$
9260 C2(1)=VAL(B$)
9270 IF C2(1)=N3 OR C2(1)>12 OR C2(1)<>INT(C2(1)) OR C2(1)<1 THEN 9130
9280 IF C8<>0 THEN 9340
9290 IF T1<3 OR T1=5 THEN 9130
9300 C2(1)=C1(T1)*(C2(1)-1)+1
9310 C2(1)=C2(1)+C9(1)-C1(T1)*INT(C9(1)/C1(T1)-0.2)-1
9320 IF C2(1)>12 THEN 9130
9330 RETURN
9340 C0(2)=28
9350 IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 9370
9360 C0(2)=29
9370 B$=SEG(R$,C7+1,C8-C7-1)
9380 IF B$<>"END" THEN 9440
9390 IF T1<3 OR T1>4 THEN 9130
9400 C2(1)=C1(T1)*C2(1)
9410 IF C2(1)>12 THEN 9130
9420 C2(2)=C0(C2(1))
9430 RETURN
9440 B$=B$&V$
```

TITLE

Gantt Chart

ABSTRACT NO:

51/07-0102/0

```
9450 C2(2)=VAL(B$)
9460 IF C2(2)=N3 OR C2(2)<>INT(C2(2)) OR C2(2)<1 THEN 9130
9470 IF C2(2)>C0(C2(1)) THEN 9130
9480 C0(2)=28
9490 RETURN
9500 PRINT "Invalid Date!GG"
9510 RETURN
9520 FOR R1=1 TO 42
9530     FOR R2=1 TO L1
9540         PRINT "ROW ";R2;"COLUMN ";R1;": ";
9550         INPUT D9(R2,R1)
9560     NEXT R2
9570 NEXT R1
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
  DISK VERSION -- GANTT1/AUTOLOAD

1 GO TO 100

3 REM ************* INPUT OF CHART CHARACTERISTICS ***** TAPE VERSION

4 FIND F1*3-1
5 GOSUB 370
6 RETURN

7 REM ************************ INPUT OF WEEKLY DATA ***** TAPE VERSION

8 FIND F1*3
9 GOSUB 370
10 RETURN

11 REM ************************ GENERATE GANTT CHART ***** TAPE VERSION

12 FIND F1*3+1
13 GO TO 90
14 RETURN

23 REM ************* INPUT OF CHART CHARACTERISTICS ***** DISK VERSION

24 P$="@GANTT1/PROGRAM/INPUT/CHART"
25 GOSUB 500
26 RETURN

27 REM ************************ INPUT OF WEEKLY DATA ***** DISK VERSION

28 P$="@GANTT1/PROGRAM/INPUT/DATA"
29 GOSUB 500
30 RETURN

31 REM ************************ GENERATE GANTT CHART ***** DISK VERSION

32 P$="@GANTT1/PROGRAM/OUTPUT/NODOC"
33 GO TO 80
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
34 RETURN
80 DELETE 101,65000
81 DELETE 1,80
82 DELETE 90,99
83 M9=MEMORY
84 APPEND P$;100
85 GO TO 100
90 DELETE 101,65000
91 DELETE 1,90
92 M9=MEMORY
93 CALL "BAPPEND",100
100 INIT
110 PRINT "L* GANTT CHARTS *JJ"
120 CALL "TIME",X$
130 IF X$<>"" THEN 300
140 PRINT "Indicate Date & Time: _________ ________"
150 PRINT "                    ex: 01-JAN-79 14:37:00HHHHHHHHHHHHHHHHHHHK";
160 INPUT X$
170 CALL "SETTIM",X$
300 PRINT "JJJ     UDK TASK"
310 PRINT "     --- ----"
315 PRINT "J          *TAPE VERSION*J"
320 PRINT "       1   DESCRIBE CHART"
330 PRINT "       2   INPUT WEEKLY DATA"
339 PRINT "       3   GENERATE CHART"
340 PRINT "J          *DISK VERSION*J"
342 PRINT "       6   DESCRIBE CHART"
344 PRINT "       7   INPUT WEEKLY DATA"
346 PRINT "       8   GENERATE CHART"
350 PRINT "JJ     The 'Generate Chart' Task will not return "
351 PRINT "      to the Menu ";
353 PRINT "(Memory Considerations)!"
355 PRINT "J     Press a Function Key!GG"
360 RETURN
370 SET NOKEY
380 CALL "BAPPEND",600
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
390 GOSUB 600
400 SET KEY
410 DELETE 601,65000
420 FOR I1=1 TO 700
430 NEXT I1
440 PRINT "L"
450 GOSUB 300
460 RETURN
500 SET NOKEY
501 APPEND P$;600
510 GOSUB 600
520 SET KEY
530 DELETE 601,65000
540 FOR I1=1 TO 700
550 NEXT I1
560 GOSUB 300
570 RETURN
600 INIT
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

DISK VERSION -- GANTT1/PROGRAM/INPUT/CHART

```
100 INIT
110 PRINT "LThis Task allows characteristics of a GANTT Chart"
120 PRINT "to be described.JJ"
130 DIM A$(923),A1(61),A2(16),A3(16),Z$(200)

140 REM ********************************* R4 - LENGTH OF DESCRIPTION AREA

150 R4=31.5

160 REM *********** A$ - CHART CHARACTERISTICS, INITIALIZE TO 923 BLANKS

170 A$="                                                              "
180 FOR I1=1 TO 17
190     A$=A$&"                                                       "
200 NEXT I1
210 A$=A$&"                          "

220 REM ****** A1 - POSITIONS IN A$ WHICH START THE CHARACTERISTICS DESC

230 DATA 1,16,31,76,92,98,128,158,166,168,170,172,173,203,233,235,237
240 DATA 238,268,298,300,302,303,333,363,365,367,368,398,428,429,459
250 DATA 460,490,491,521,522,552,553,583,584,614,615,645,646,676,677
260 DATA 707,708,738,739,769,770,900,801,831,832,862,863,893,894
270 RESTORE 230
280 READ A1

290 REM ***** A2 & A3 - ROWS HAVING SINGLE AND DOUBLE VERT DESC'S, INIT

300 FOR I1=1 TO 16
310     A2(I1)=0
320     A3(I1)=0
330 NEXT I1

340 REM ********************* PROMPT FOR CHARACTERISTICS AND STORE IN A$

350 PRINT "  PROJECT NAME: _____ "
```

TITLE	ABSTRACT NO:
GANTT CHART	51/07-0102/0

```
360 PRINT "                    _________________"
370 PRINT "                 KK";
380 INPUT X$
390 PRINT "                 ";
400 INPUT Y$
410 IF LEN(X$)<>0 THEN 450
420 PRINT "                 type Project Name on Line 1 and optionally";
430 PRINT " on Line 2!GG"
440 GO TO 350
450 IF LEN(X$)<=15 AND LEN(Y$)<=15 THEN 480
460 PRINT "                 15 characters maximum for each Line!GG"
470 GO TO 350
480 A$=REP(X$,A1(1),LEN(X$))
490 IF LEN(Y$)=0 THEN 510
500 A$=REP(Y$,A1(2),LEN(Y$))
510 PRINT "   CHART TITLE: _____________________________________________";
520 PRINT "_____"
530 PRINT "                 K";
540 INPUT X$
550 IF LEN(X$)<=45 AND LEN(X$)<>0 THEN 580
560 PRINT "                 45 characters maximum!GG"
570 GO TO 510
580 A$=REP(X$,A1(3),LEN(X$))
590 PRINT "          CODE: ________________HHHHHHHHHHHHHHHH";
600 INPUT X$
610 IF LEN(X$)<=16 THEN 640
620 PRINT "                 16 characters maximum!GG"
630 GO TO 590
640 A$=REP(X$,A1(4),LEN(X$))
650 PRINT "PROJECT NUMBER: _____HHHHHH";
660 INPUT X$
670 IF LEN(X$)<=6 AND LEN(X$)<>0 THEN 700
680 PRINT "                 6 characters maximum!GG"
690 GO TO 650
700 A$=REP(X$,A1(5),LEN(X$))
710 PRINT "     SIGNATURE: ___________________________________"
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
720 PRINT "                                                    "
730 PRINT "                KK";
740 INPUT X$
750 PRINT "                ";
760 INPUT Y$
770 IF LEN(X$)<>0 THEN 810
780 PRINT "                type Signature on Line 1 and optionally ";
790 PRINT "on Line 2!GG"
800 GO TO 710
810 IF LEN(X$)<=30 AND LEN(Y$)<=30 THEN 840
820 PRINT "                30 characters maximum for each Line!GG"
830 GO TO 710
840 A$=REP(X$,A1(6),LEN(X$))
850 IF LEN(Y$)=0 THEN 870
860 A$=REP(Y$,A1(7),LEN(Y$))
870 FOR I1=1 TO 300
880 NEXT I1
890 L4=5
900 L1=0
910 PRINT "LVERTICAL DESCRIPTIONS (4 MAX)J"
920 L4=L4+5
930 L1=L1+1
940 IF L1>4 THEN 1720
950 PRINT "J"
960 PRINT " SINGLE or DOUBLE: _ (type '0' when done)"
970 PRINT "                   K";
980 INPUT W$
990 IF W$="0" OR W$="S" OR W$="D" THEN 1020
1000 PRINT "                  type '0','S', or 'D'!GG"
1010 GO TO 960
1020 IF W$="0" THEN 1720
1030 PRINT "          BEG ROW:  HH";
1040 INPUT X$
1050 IF LEN(X$)=1 OR LEN(X$)=2 THEN 1080
1060 PRINT "                  type 1 or 2 numbers please!GG"
1070 GO TO 1030
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1080 GOSUB 2720
1090 IF J1=0 THEN 1120
1100 PRINT "                    type numbers only!GG"
1110 GO TO 1030
1120 J1=VAL(X$)
1130 IF J1=>1 AND J1<=16 THEN 1170
1140 PRINT "                    Vertical Description can span ";
1150 PRINT "Rows 1 thru 16 only!GG"
1160 GO TO 1030
1170 Y$=X$
1180 PRINT "          END ROW:  HH";
1190 INPUT X$
1200 IF LEN(X$)=1 OR LEN(X$)=2 THEN 1230
1210 PRINT "                1 or 2 numbers only!GG"
1220 GO TO 1180
1230 J2=VAL(X$)
1240 IF J2>J1 AND J2<=16 THEN 1290
1250 PRINT "                    Ending Row must be greater than Beg Row"
1260 PRINT "                     and less then Row 16!GG"
1270 GO TO 1030

1280 REM ****************************** CHECK IF ROWS ARE ALREADY SPANNED

1290 J3=0
1300 FOR I1=J1 TO J2
1310     IF A2(I1)<>0 THEN 1330
1320     IF A3(I1)=0 THEN 1350
1330     J3=1
1340     I1=J2
1350 NEXT I1
1360 IF J3=0 THEN 1400
1370 PRINT "                    a Vert Desc already spans this Row!GG"
1380 GO TO 960

1390 REM ******** STORE SPANNED ROWS IN SINGLE & DOUBLE ARRAYS,. A2 & A3
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
1400 FOR I1=J1 TO J2
1410     IF W$="D" THEN 1440
1420     A2(I1)=1
1430     GO TO 1450
1440     A3(I1)=1
1450 NEXT I1
1460 A$=REP(Y$,A1(L4),LEN(Y$))
1470 A$=REP(X$,A1(L4+1),LEN(X$))
1480 A$=REP(W$,A1(L4+2),1)

1490 REM ********** L3 - NO OF SPACES AVAILABLE FOR VERTICAL DESCRIPTION

1500 L3=(J2-J1+1)*2+(J2-J1)
1510 IF W$="D" THEN 1540
1520 PRINT "      DESCRIPTION: ";
1530 GO TO 1550
1540 PRINT "   DESCRIPTION #1: ";
1550 GOSUB 2810
1560 GOSUB 2860
1570 INPUT X$
1580 IF LEN(X$)<=L3 AND LEN(X$)<>0 THEN 1610
1590 PRINT "                    ";L3;" characters maximum!GG"
1600 GO TO 1540
1610 A$=REP(X$,A1(L4+3),LEN(X$))
1620 IF W$="S" THEN 920
1630 PRINT "   DESCRIPTION #2: ";
1640 GOSUB 2810
1650 GOSUB 2860
1660 INPUT X$
1670 IF LEN(X$)<=L3 AND LEN(X$)<>0 THEN 1700
1680 PRINT "                    ";L3;" characters only!GG"
1690 GO TO 1630
1700 A$=REP(X$,A1(L4+4),LEN(X$))
1710 GO TO 920
1720 FOR I1=1 TO 300
1730 NEXT I1
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1740 L4=28
1750 L1=0
1760 PRINT "LHORIZONTAL DESCRIPTIONSJ"
1770 L4=L4+2
1780 L1=L1+1
1790 IF L1>16 THEN 2042
1800 PRINT "J    DESCRIPTION: "
1810 PRINT "                  type '0' when complete"
1820 IF A2(L1)=0 THEN 1850
1830 L3=R4-4
1840 GO TO 1890
1850 IF A3(L1)=0 THEN 1880
1860 L3=R4-6
1870 GO TO 1890
1880 L3=R4-1
1890 PRINT "                KK";
1900 GOSUB 2810
1910 GOSUB 2860
1920 INPUT X$
1930 IF X$="0" THEN 2069
1940 IF LEN(X$)<=L3 AND LEN(X$)<>0 THEN 1970
1950 PRINT "J                ";INT(L3);" characters maximum!GG"
1960 GO TO 1800
1970 PRINT "J          LINE?  H";
1980 INPUT Y$
1990 IF Y$="Y" OR Y$="N" THEN 2020
2000 PRINT "              type a 'Y' or a 'N'!GG"
2010 GO TO 1970
2020 A$=REP(Y$,A1(L4),1)
2030 A$=REP(X$,A1(L4+1),LEN(X$))
2040 GO TO 1770

2041 REM ********************************** CHECK THAT # OF ROWS IS OKAY

2042 IF L1=>16 THEN 2056
2043 J1=0
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2044 FOR I1=L1 TO 16
2045     IF A2(I1)<>0 THEN 2047
2046     IF A3(I1)=0 THEN 2049
2047     J1=1
2048     I1=16
2049 NEXT I1
2050 IF J1=0 THEN 2056
2051 PRINT "JVertical Descriptions can not exceed Horizontal ";
2052 PRINT "Descriptions.  Start over!GG"
2053 GO TO 170

2055 REM ****************************************** STORE NO OF ROWS (L1)

2056 L1=L1-1
2057 X$=STR(L1)
2058 X$=REP("",1,1)
2059 A$=REP(X$,A1(9),LEN(X$))
2061 L1=L1+1
2062 FOR I1=1 TO 500
2063 NEXT I1

2068 REM ************************************* PREPARE FOR DISK STORAGE

2069 PRINT "L"
2070 CALL "UNIT",0
2080 PRINT "JJWhat shall be the name of this Chart? __________";
2090 PRINT "HHHHHHHHHH";
2100 INPUT X$
2110 IF LEN(X$)<=10 AND LEN(X$)<>0 THEN 2150
2120 PRINT "Chart name can be a maximum of 10 characters!GG"
2130 GO TO 2080
2140 INPUT X$
2150 Z$=SEG(X$,1,1)
2160 IF Z$=>"A" AND Z$<="Z" THEN 2190
2170 PRINT "Chart name must begin with an Alphabetic character!GG"
2180 GO TO 2080
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2190 FOR I1=2 TO LEN(X$)
2200     Z$=SEG(X$,I1,1)
2210     IF Z$=>"A" AND Z$<="Z" OR (Z$=>"0" AND Z$<="9") THEN 2240
2220     PRINT "No Blanks or Special Characters allowed for Chart name!GG"
2230     GO TO 2080
2240     P$=X$
2250     PRINT "JInsert in Unit #0 the Disk to contain Chart ";X$
2260     PRINT "To continue, type <CR>. ";
2270     INPUT Y$
2280     CALL "MOUNT",0,Z$
2290     Z$=SEG(Z$,73,10)
2300     I1=VAL(Z$)
2310     IF I1>5000 THEN 2360
2320     PRINT "JThere is not enough room on this Disk to store Chart ";X$
2330     PRINT "Insert another Disk to store this Chart."
2340     GO TO 2260

2350     REM ****************************** STORE CHART CHARACTERISTICS, A$

2360     Z$="@GANTT1/"&X$
2370     X$=Z$&"/CHART"
2380     Y$=Z$&"/DATA"
2390     CALL "FILE",0,X$,Z$
2400     IF Z$="" THEN 2470
2410     PRINT "J";P$;" already exists as a Chart name.  Should it be "
2420     PRINT "over-written? _H";
2430     INPUT Z$
2440     IF Z$="N" THEN 2080
2450     IF Z$<>"Y" THEN 2410

2451     REM ************* CHECK THAT # OF ROWS IN OLD AND NEW CHART MATCH

2452     OPEN X$;1,"F",Z$
2453     READ #1:Z$
2454     Z$=SEG(Z$,A1(9),2)
2455     L2=VAL(Z$)
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2456     CLOSE 1
2457     IF L1-1=L2 THEN 2460
2458     PRINT "JTo properly access the associated Data File, this Chart ";
2459     PRINT "must have       ";L2;" Rows.  Start over!GGJJJ"
2460     FOR I1=1 TO 500
2461        FOR I1=1 TO 500
2462           GO TO 170
2469           GO TO 2480
2470           CREATE X$;1000,0
2480           OPEN X$;1,"F",Z$
2490           Q$=STR(L1)
2491           IF LEN(Q$)=2 THEN 2494
2492           Q$=REP("",1,1)
2494           A$=REP(Q$,A1(9),2)
2499           WRITE #1:A$
2500           CLOSE 1

2510           REM ********* IF DATA FILE ALREADY EXISTS THEN THIS TASK HAS BEEN
2520           REM *********  USED FOR MODIFICATION

2530           CALL "FILE",0,Y$,Z$
2540           IF Z$="" THEN 2620
2550           PRINT "JA Data File already exists for Chart ";P$;".  Should it"
2560           PRINT "be initialized? _H";
2570           INPUT Z$
2580           IF Z$="N" THEN 2690
2590           IF Z$<>"Y" THEN 2550
2600           GO TO 2630

2610           REM ************************************** SET UP DUMBY DATA FILE

2620           CREATE Y$;1000,0
2630           OPEN Y$;1,"F",Z$
2640           DELETE D9
2650           DIM D9(L1-1,42)
2660           D9=-1
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
_670            WRITE #1:"01/01/50",L1-1,D9
2680            CLOSE 1
2690            PRINT "JJJJ";P$;" Chart format stored!GG"
2700            RETURN

2710            REM ***************** SUBROUTINE - ALL MEMBERS OF X$ MUST BE NUMERI

2720            J1=0
2730            FOR K1=1 TO LEN(X$)
2740                Z$=SEG(X$,K1,1)
2750                IF Z$=>"0" AND Z$<="9" THEN 2780
2760                J1=1
2770                K1=LEN(X$)
2780            NEXT K1
2790            RETURN

2800            REM ******************** SUBROUTINE - PRINT L3 UNDERLINE CHARACTERS

2810            FOR K1=1 TO L3
2820                PRINT "_";
2830            NEXT K1
2840            RETURN

2850            REM ******************** SUBROUTINE - PRINT L3 BACKSPACE CHARACTERS

2860            FOR K1=1 TO L3
2870                PRINT "H";
2880            NEXT K1
2890            RETURN
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

DISK VERSION -- GANTT1/PROGRAM/INPUT/DATA

```
100 INIT
110 PRINT "LThis Task allows Data Editing."
120 DIM H5(1,28),C1(4),C2(3),C3(3),C9(3),F$(51),G$(54),R$(24)
130 DIM W$(48),B2(3),B9(3),C$(9),C0(12),D2(3),V$(15),M$(1)
140 M$=CHR(13)

150 REM ***************** INITIALIZATION OF CALENDAR FUNCTION VARIABLES

160 DATA 31,28,31,30,31,30,31,31,30,31,30,31,1,7,1,3
170 RESTORE 160
180 READ C0,C1
190 F$="   JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC "
200 N3=-2.713E+288
210 V$=STR(N3)
220 W$="          SUN   MON  TUESWEDNES THURS   FRI SATUR"
230 G$="           DAYS      WEEKD     MONTHS    QUARTERS YEARS"

240 REM ********* READ FROM DISK: R$ - DATE, L1 - # OF ROWS, D9 - DATA

250 PRINT "JJWhich Chart? _____HHHHHHHHHH";
260 INPUT X$
270 IF LEN(X$)<=10 AND LEN(X$)<>0 THEN 300
280 PRINT "Chart name must be less than 10 characters!GG"
290 GO TO 250
300 Z$=SEG(X$,1,1)
310 IF Z$=>"A" AND Z$<="Z" THEN 340
320 PRINT "Chart name must begin with an Alphabetic character!GG"
330 GO TO 250
340 FOR I1=2 TO LEN(X$)
350     Z$=SEG(X$,I1,1)
360     IF Z$=>"A" AND Z$<="Z" OR (Z$=>"0" AND Z$<="9") THEN 390
370     PRINT "No Blanks or Special characters allowed for Chart name!GG"
380     GO TO 250
390     O$=X$
400     CALL "UNIT",0
410     PRINT "JInsert in Unit #0 the Disk containing Chart ";X$
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
420     PRINT "To continue, type <CR>. ";
430     INPUT Y$
440     Z$="@GANTT1/"&X$
450     Z$=Z$&"/DATA"
460     Q$=Z$
470     CALL "MOUNT",0,Y$
480     CALL "FILE",0,Z$,Y$
490     IF Y$<>"" THEN 520
500     PRINT "Chart ";X$;" is not stored on this Disk!GG"
510     GO TO 410
520     OPEN Z$;1,"F",Y$
530     READ #1:R$,L1
540     DIM D9(L1,28)
550     READ #1:D9
560     CLOSE 1

570     REM ********************** P$ & R$ - DATE FOR FIRST COLUMN OF DATA

580     Z$=R$
590     PRINT "JJJBeginning Data Date is ";R$;".  Is this correct? ";
600     INPUT Y$
610     GOSUB 3380
620     P$=R$
630     IF Y$="Y" THEN 770
640     IF Y$<>"N" THEN 590
650     PRINT "JNew Date: _________HHHHHHHH";
660     INPUT R$
670     GOSUB 3380
680     IF H1<>0 THEN 650
690     GOSUB 3040
700     B$=SEG(W$,6*C7+1,6)
710     IF B$="   SUN" THEN 740
720     PRINT R$;" is a ";B$;".  Indicate a Sunday!GG"
730     GO TO 650
740     P$=R$
750     GOSUB 2390
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
760     REM ************************************************** DATA EDITING

770     PRINT "LDuring Data modification, type:J"
780     PRINT "  <CR> if present value is correct;"
790     PRINT "  -1 if value is not to print; or "
800     PRINT "  0 if it is to be calculated."
810     PRINT "JJJTyping a '0' for any of the following Questions will ";
820     PRINT "cause              an exit from the indicated Process."
830     R$=P$
840     PRINT "JJDo you want to go thru each Week(1), only selected ";
850     PRINT "Weeks(2),            or Neither(0)? _H";
860     INPUT Y$
870     IF Y$="0" THEN 1820
880     IF Y$="1" THEN 1130
890     IF Y$<>"2" THEN 840
900     R$=P$
910     GOSUB 3380
920     B2=C2
930     C3=C2
940     GOSUB 3380
950     GOSUB 3040
960     Z1=C8
970     PRINT "JWhich Date: _______HHHHHHHH";
980     INPUT R$
990     IF R$="0" THEN 770
1000     GOSUB 3380
1010     IF H1<>0 THEN 970
1020     B2=C2
1030     C3=C2
1040     GOSUB 3040
1050     Z1=C8-Z1
1060     IF Z1=>0 AND Z1<=27 THEN 1090
1070     PRINT R$;" is out of the 28-Week range!GG"
1080     GO TO 900
1090     I1=Z1+1
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
.100     GOSUB 1290
1110     GO TO 900

1120     REM ****************************************** LOOP THRU EACH COLUMN

1130     C5=0
1140     R$=P$
1150     GOSUB 3380
1160     FOR I1=1 TO 28

1170         REM ************************************* FIND DATE FOR THIS COLUMN

1180         GOSUB 2720
1190         PRINT M$;"JJ"
1200         PRINT "Should Data for ";C2(1);"/";C2(2);"/";C2(3);" be Modified?";
1210         PRINT " H";
1220         INPUT Y$
1230         IF Y$="0" THEN 1760
1240         IF Y$="N" THEN 1740
1250         IF Y$<>"Y" THEN 1200
1260         GOSUB 1290
1270         GO TO 1740

1280         REM ********************************************** LOOP THRU EACH ROW

1290         FOR I2=1 TO L1
1300             Y$=STR(D9(I2,I1))
1310             Y$=REP("",1,1)
1320             J1=POS(Y$,".",1)
1330             IF J1=0 THEN 1390
1340             Y$=REP("/",J1,1)
1350             Y$=REP("",LEN(Y$)-1,2)
1360             IMAGE 6A,2D,2A,6A,15A
1370             IF Y$<>"1/0E" THEN 1390
1380             Y$="0/00"
1390             PRINT USING 1360:"  Row ";I2;": ";Y$;"        HHHHHH"
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1400            PRINT "                          K";
1410            INPUT X$
1415            IF X$="-1" THEN 1700
1420            IF LEN(X$)<=6 THEN 1450
1430            PRINT "                          6 Characters maximum!GG"
1440            GO TO 1390
1450            IF LEN(X$)=0 THEN 1720

1460            REM *** SUBROUTINE - ALL CHARACTERS OF X$ MUST BE NUMERIC OR A "/

1470            J1=0
1480            FOR K1=1 TO LEN(X$)
1490                Z$=SEG(X$,K1,1)
1500                IF Z$=>"0" AND Z$<="9" OR Z$="/" THEN 1530
1510                J1=1
1520                K1=LEN(X$)
1530            NEXT K1
1540            IF J1=0 THEN 1580
1550            PRINT "                          Type only numbers and, optionally, ";
1560            PRINT " a '/'!GG"
1570            GO TO 1390
1580            J1=POS(X$,"/",1)
1590            IF J1=0 THEN 1700
1600            IF J1+3>LEN(X$) THEN 1650
1610            Z$=SEG(X$,J1+3,1)
1620            IF Z$=" " THEN 1650
1630            PRINT "Only two numbers allowed after the '/'!GG"
1640            GO TO 1390
1650            X$=REP(".",J1,1)
1660            U1=VAL(X$)
1670            U2=1.0E-4
1680            U1=U1+U2
1690            GO TO 1710
1700            U1=VAL(X$)
1710            D9(I2,I1)=U1
1720        NEXT I2
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1730        RETURN
1740        C5=1
1750        GO TO 1770
1760        I1=42
1770    NEXT I1
1780    GO TO 770

1790    REM ********************************************* TOTAL CALCULATIONS
1800    REM *** L2 - NO OF ROWS TO BE ADDED TOGETHER * L3 - # OF TOTAL ROW
1810    REM * H2 - ARRAY OF ROW #s TO BE ADDED * H5 - ARRAY OF SUMMED DATA

1820    PRINT "LTotal Calculations"
1830    PRINT "JJHow many Rows? __ (type '0' when complete)"
1840    PRINT "                  K";
1850    INPUT L2
1860    IF L2<=0 THEN 2260
1870    IF L2=>1 AND L2<=L1 THEN 1900
1880    PRINT "Only ";L1;" Rows exist!GG"
1890    GO TO 1830
1900    DELETE H2
1910    DIM H2(L2)
1920    PRINT "JIndicate Rows: ____________________________________________"
1930    PRINT "                  K";
1940    INPUT H2
1950    PRINT "JIndicate Total Row #: __HH";
1960    INPUT L3
1970    IF L3=>1 AND L3<=L1 THEN 1990
1980    GO TO 1880
1990    J1=0
2000    FOR I1=1 TO L2
2010        IF H2(I1)=>1 AND H2(I1)<=L1 THEN 2060
2020        PRINT "Row #";H2(I1);" is outside Row Range.  Row #'s must ";
2030        PRINT "fall between 1 and ";L1;"!GG"
2040        J1=1
2050        GO TO 2100
2060        IF H2(I1)<>L3 THEN 2100
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
2070        PRINT "Row #";H2(I1);" to be Added is the same as the Total Ro: ";
2080        PRINT "#";L3;".                          They must be different!GG"
2090        J1=1
2100    NEXT I1
2110    IF J1=1 THEN 1830
2120    H5=0

2130    REM ************** ADD 1 ROW AT A TIME ACROSS ALL 28 DATA COLUMNS

2140    FOR I1=1 TO L2
2150        FOR I2=1 TO 28
2160            IF D9(H2(I1),I2)<0 THEN 2190
2170            H5(1,I2)=H5(1,I2)+D9(H2(I1),I2)
2180            GO TO 2200
2190            H5(1,I2)=-1
2200        NEXT I2
2210    NEXT I1
2220    FOR I1=1 TO 28
2230        D9(L3,I1)=H5(1,I1)
2240    NEXT I1
2250    GO TO 1820
2260    GOSUB 2390
2270    PRINT "LWould you like to go thru the Editing process again? _H";
2280    INPUT Y$
2290    IF Y$="Y" THEN 580
2300    IF Y$<>"N" THEN 2270
2310    OPEN Q$;1,"F",Z$
2320    WRITE #1:P$,L1,D9
2330    CLOSE 1
2340    PRINT "JJBeginning Date, Number of Rows, and the Weekly Data ";
2350    PRINT "has been            stored in the Data File for Chart ";
2360    PRINT O$;"!GG"
2370    RETURN

2380    REM ************************************** SUBROUTINE - LIST DATA
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2390     PRINT "LWould you like a Print-out of the existing Data? H";
2400     INPUT Y$
2410     IF Y$="N" THEN 2660
2420     IF Y$<>"Y" THEN 2390
2430     R$=P$

2440     REM ************************************** LOOP THRU EACH COLUMN

2450     C5=0
2460     FOR I1=1 TO 28

2470         REM ********************************** FIND DATE FOR THIS COLUMN

2480         GOSUB 3380
2490         GOSUB 2720
2500         PRINT M$;"JJCOLUMN: ";I1;"  (";C2(1);"/";C2(2);"/";C2(3);")"

2510         REM ********************************************* LOOP THRU EACH ROW

2520         FOR I2=1 TO L1
2530             Y$=STR(D9(I2,I1))
2540             IF Y$<>" 1.0E-4" THEN 2570
2550             Y$="0/0"
2560             GO TO 2620
2570             Y$=REP("",1,1)
2580             I3=POS(Y$,".",1)
2590             IF I3=0 THEN 2620
2600             Y$=REP("/",I3,1)
2610             Y$=REP("",LEN(Y$)-1,2)
2620             PRINT "  R";I2;": ";Y$;"  ";
2630         NEXT I2
2640         C5=C5+1
2650     NEXT I1
2660     PRINT M$;"JJJTo continue, type <CR>. ";
2670     INPUT Y$
2680     RETURN
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2690     REM ******************** SUBROUTINE - DATE + OR - C5 TIME SEGMENTS
2700     REM ************************ C2=(mm,dd,yy)  C5=# of time segments
2710     REM ********************************** T1=time mesh * C6,C7=scratch

2720     IF C5=0 THEN 2820
2730     C6=SGN(C5)
2740     T1=2
2750     GO TO T1 OF 2760,2760,2950,2950,2970
2760     C2(2)=C2(2)+C1(T1)*C5
2770     C0(2)=28
2780     IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 2800
2790     C0(2)=29
2800     IF C2(2)<1 OR C2(2)>C0(C2(1)) THEN 2830
2810     C0(2)=28
2820     RETURN
2830     IF C6+1 THEN 2920
2840     C2(1)=C2(1)+C6
2850     IF C2(1)=>1 AND C2(1)<=12 THEN 2900
2860     C2(3)=C2(3)+C6
2870     C2(1)=C2(1)-12*C6
2880     IF T1>2 THEN 2850
2890     GO TO 2910
2900     IF T1>2 THEN 2990
2910     IF C6+1 THEN 2770
2920     C2(2)=C2(2)-C6*C0(C2(1))
2930     IF C6+1 THEN 2840
2940     GO TO 2770
2950     C2(1)=C2(1)+C1(T1)*C5
2960     GO TO 2850
2970     C2(3)=C2(3)+C5
2980     GO TO 2810
2990     C2(2)=C2(2) MIN C0(C2(1))
3000     GO TO 2810

3010     REM *************** SUBROUTINE - DAY #, WEEK #, & DAY OF THE WEEK
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
3020     REM ************************ C4=day # * C8=week # * C7=day of week
3030     REM ********************************** C2=(mm,dd,yy) * B3=scratch

3040     C4=0.25*C2(3)=INT(0.25*C2(3)) AND C2(1)>2
3050     IF C2(1)=1 THEN 3090
3060     FOR B3=1 TO C2(1)-1
3070         C4=C4+C0(B3)
3080     NEXT B3
3090     C4=C4+C2(2)+365*C2(3)+INT(0.25*(C2(3)-1))
3100     C8=INT(C4/7)+1
3110     C7=C4-(C8-1)*7+1
3120     RETURN

3130     REM ********************* SUBROUTINE - TIME SEGMENTS BETWEEN DATES
3140     REM ************************************************** T1=time mesh
3150     REM *************** 1st date=B2=(mm,dd,yy) * c8=# of time segments
3160     REM **** 2nd date=C3=(mm,dd,yy) * B2,B3,C0,C2,C4,C5,C6,C7=scratch

3170     T1=2
3180     GO TO T1 OF 3190,3190,3300,3300,3330
3190     C2=C3
3200     GOSUB 3040
3210     C5=C4
3220     C6=C8
3230     C2=B2
3240     GOSUB 3040
3250     IF T1=2 THEN 3280
3260     C8=C5-C4
3270     RETURN
3280     C8=C6-C8
3290     RETURN
3300     C8=INT((C3(1)+2)/C1(T1))-INT((B2(1)+2)/C1(T1))
3310     C8=12/C1(T1)*(C3(3)-B2(3))+C8
3320     RETURN
3330     C8=C3(3)-B2(3)
3340     RETURN
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
3350     REM ********************************* SUBROUTINE - DATE ENTRY VERIFY
3360     REM ********************** H1=flag (118=bad 0=OK) * R$=date string
3370     REM ******* C2=(mm,dd,yy) * C7=pos of 1st "/" * C8=pos of 2nd "/"

3380     H1=0
3390     T1=2
3400     C7=POS(R$,"/",1)
3410     C8=POS(R$,"/",C7+1)
3420     C2(2)=1
3430     B$=SEG(R$,(C8 MAX C7)+1,LEN(R$)-(C8 MAX C7))
3440     B$=B$&V$
3450     C2(3)=VAL(B$)
3460     IF C2(3)<>N3 AND C2(3)=>1 AND C2(3)=INT(C2(3)) THEN 3490
3470     H1=118
3480     GO TO 3840
3490     IF C7<>0 THEN 3530
3500     IF T1<>5 THEN 3470
3510     C2(1)=1
3520     RETURN
3530     B$=SEG(R$,1,C7-1)
3540     IF B$<>"END" THEN 3590
3550     IF T1<>5 AND C8<>0 THEN 3470
3560     C2(1)=12
3570     C2(2)=31
3580     RETURN
3590     B$=B$&V$
3600     C2(1)=VAL(B$)
3610     IF C2(1)=N3 OR C2(1)>12 OR C2(1)<>INT(C2(1)) OR C2(1)<1 THEN 3470
3620     IF C8<>0 THEN 3680
3630     IF T1<3 OR T1=5 THEN 3470
3640     C2(1)=C1(T1)*(C2(1)-1)+1
3650     C2(1)=C2(1)+C9(1)-C1(T1)*INT(C9(1)/C1(T1)-0.2)-1
3660     IF C2(1)>12 THEN 3470
3670     RETURN
3680     C0(2)=28
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
3690     IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 3710
3700     C0(2)=29
3710     B$=SEG(R$,C7+1,C8-C7-1)
3720     IF B$<>"END" THEN 3780
3730     IF T1<3 OR T1>4 THEN 3470
3740     C2(1)=C1(T1)*C2(1)
3750     IF C2(1)>12 THEN 3470
3760     C2(2)=C0(C2(1))
3770     RETURN
3780     B$=B$&V$
3790     C2(2)=VAL(B$)
3800     IF C2(2)=N3 OR C2(2)<>INT(C2(2)) OR C2(2)<1 THEN 3470
3810     IF C2(2)>C0(C2(1)) THEN 3470
3820     C0(2)=28
3830     RETURN
3840     PRINT "Invalid Date!GG"
3850     RETURN
3860     FOR R1=1 TO 42
3870         FOR R2=1 TO L1
3880             PRINT "ROW ";R2;"COLUMN ";R1;": ";
3890             INPUT D9(R2,R1)
3900         NEXT R2
3910     NEXT R1
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

DISK VERSION -- GANTT1/PROGRAM/OUTPUT/NODOC

```
100 INIT
110 PRINT "LThis Task generates GANTT Chart"
560 M9=MEMORY
570 DIM A1(61),A2(16),A3(16),A4(12),A5(12),K$(300),S(321)
580 DIM C1(4),C2(3),C3(3),C9(3),F$(51),G$(54),R$(24),V$(15),W$(48)
590 DIM B2(3),B9(3),C$(9),C0(12),D2(3),D5(3),H2(3)
600 DATA 31,28,31,30,31,30,31,31,30,31,30,31,1,7,1,3
610 N6=0
620 RESTORE 600
630 READ C0,C1
650 F$="   JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC "
660 N3=-2.713E+288
670 V$=STR(N3)
680 W$="          SUN   MON  TUESWEDNES THURS   FRI SATUR"
690 G$="           DAYS      WEEKS     MONTHS    QUARTERS YEARS"

710 REM .....strokes.....

720 DATA 0,0,0,7,4,7,5,5.25,4,3.5,5,1.75,4,0,0,0,2.5,7,5,0,4,3,1,3
730 DATA 2,0,2,1,3,1,3,0,2,0,5,0,0,0,0,3.5,2,3.5,0,3.5,0,7,5,7,4,7,4,2
740 DATA 3,0,1,0,0,1,5,7,0,7,0,0,5,0,5,3.5,3,3.5,0,0,0,7,3,7,5,5,5,2,3
750 DATA 0,0,0,7,0,3.5,5,3.5,5,7,5,0,0,0,0,7,5,7,5,0,3,2
760 DATA 2.5,0,2.5,7,0,7,5,7,0,0,5,0,5,3.5,0,3.5,0,7,5,7,5,4,0,0,5,0
770 DATA 0,5,7,2.5,3.5,0,7,5,0,0,0,7,2.5,0,5,7,5,0,5,7,0,7,0,3.5,5,3.5
780 DATA 5,7,0,7,0,0,5,0,5,3.5,0,3.5,0,0,0,7,5,7,5,3.5,0,3.5,3,3.5,5,0
790 DATA 0,0,7,5,0,5,7,1,0,2.5,0,2.5,7,1,7,4,7,2.5,7,2.5,0,4,0
800 DATA 0,0,3.5,5,7,0,3.5,5,0,0,3.5,0,7,0,0,5,0,5,7,0,7,0,0,2.5,3.5,5
810 DATA 0,5,7,3,0,3,7,0,2,5,2

820 REM .....begins.....

830 DATA 25,0,1,3,15,35,37,45,59,61,59,71,82,92,92,92,104,108,108
840 DATA 112,112,120,129,129,138,140,146,158,154,168,168,181,189,191
850 DATA 189,197,204,216,224,234

860 REM .....ends.....
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
870 DATA 34,0,16,16,24,48,48,58,66,66,70,84,93,101,101,103,111,115,113
880 DATA 123,123,129,138,136,147,145,155,167,167,177,181,188,196,194
890 DATA 204,204,217,223,233,241
900 RESTORE 720
910 READ S
930 G1=0
940 G2=0
950 G3=130
960 G4=100
990 G5=5
1000 G6=3
1010 G7=3
1020 G8=0.5
1050 R1=3
1060 R2=5
1070 R3=3
1080 R4=31.5
1130 E1=1.7
1140 E2=2.5
1150 E3=1.825
1160 E4=3.5
1170 E5=1
1180 E6=2.5
1190 E7=1
1200 E8=2.5
1210 E9=0.866
1220 E0=1.5
1240 DATA 1,16,31,76,92,98,128,158,166,168,170,172,173,203,233,235
1250 DATA 237,238,268,298,300,302,303,333,363,365,367,368,398,428,429
1260 DATA 459,460,490,491,521,522,552,553,583,584,614,615,645,646,676
1270 DATA 677,707,708,738,739,769,770,800,801,831,832,862,863,893,894
1280 RESTORE 1240
1290 READ A1
1310 FOR I1=1 TO 16
1320     A2(I1)=0
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1330     A3(I1)=0
1340 NEXT I1
1360 CALL "UNIT",0
1370 PRINT "JJWhat is the Name of the GANTT Chart? _________"
1380 PRINT "                                        K";
1390 INPUT P$
1400 PRINT "JJInsert in Unit #0 the Disk containing Chart ";P$
1410 PRINT "To continue, type <CR>. ";
1420 INPUT Y$
1430 CALL "MOUNT",0,Y$
1440 Z$="@GANTT1/"&P$
1445 N$=Z$&"/DATA"
1450 X$=Z$&"/CHART"
1460 CALL "FILE",0,X$,Y$
1470 IF Y$<>"" THEN 1500
1480 PRINT "J";X$;" Chart does not exist on this Disk!GG"
1490 GO TO 1370
1500 OPEN X$;1,"R",Y$
1510 DELETE 480,1500
1520 DIM A$(923)
1530 READ #1:A$
1540 CLOSE 1
1570 N$=Z$&"/DATA"
1580 OPEN N$;1,"R",Y$
1590 READ #1:R$,L1
1600 DELETE 1400,1590
1610 M9=MEMORY
1620 READ #1:D9
1630 CLOSE 1
1670 GOSUB 9050
1680 GOSUB 8720
1690 Z1=C8
1700 Z$=R$
1710 PRINT "JJBeginning Data Date is ";Z$
1720 PRINT "Indicate beginning Column Date: _______HHHHHHHH";
1730 INPUT R$
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
1740 GOSUB 9050
1750 IF H1 THEN 1720
1760 GOSUB 8720
1770 Z1=C8-Z1
1780 IF Z1=>0 AND Z1<=27 THEN 1830
1790 PRINT "Beginning Date must be within 28 weeks of ";Z$
1800 PRINT R$;" is ";Z1;" weeks from ";Z$
1810 R$=Z$
1820 GO TO 1670
1830 B$=SEG(W$,6*C7+1,6)
1840 IF B$="   SUN" THEN 1880
1850 PRINT R$;" is ";B$;". Select a Sunday!GG"
1860 R$=Z$
1870 GO TO 1670
1880 N5=Z1
1890 I2=0
1910 PRINT "LRow Descriptions for Chart ";P$;"J"
1920 FOR I1=30 TO 61 STEP 2
1930     I2=I2+1
1940     Y$=SEG(A$,A1(I1),1)
1950     IF Y$=" " THEN 2000
1960     X$=SEG(A$,A1(I1+1),30)
1970     PRINT "  ROW ";I2;": ";X$
1980     GO TO 2000
2000 NEXT I1
2010 PRINT "JJIndicate the 3 Rows to be Graphed: _______HHHHHHHH"
2020 PRINT "(If necessary, type 0's to make 3) K";
2030 INPUT H2
2040 I2=0
2050 FOR I1=1 TO 3
2060     IF H2(I1)=0 THEN 2100
2070     IF H2(I1)=>1 AND H2(I1)<=L1+1 THEN 2100
2080     I1=3
2090     I2=1
2100 NEXT I1
2110 IF I2=0 THEN 2150
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2120 PRINT "Rows must be between 1 and ";L1;"!GG"
2130 GO TO 2010
2150 PRINT "LLegend name for:J"
2160 IF H2(1)=0 THEN 2220
2170 PRINT "  Row ";H2(1);": _____HHHHHHHHHHHHHHH";
2180 INPUT O$
2190 IF LEN(O$)<=15 AND LEN(O$)<>0 THEN 2220
2200 PRINT "          15 characters maximum!GG"
2210 GO TO 2170
2220 IF H2(2)=0 THEN 2280
2230 PRINT "  Row ";H2(2);": _____HHHHHHHHHHHHHHH";
2240 INPUT P$
2250 IF LEN(P$)<=15 AND LEN(P$)<>0 THEN 2280
2260 PRINT "          15 characters maximum!GG"
2270 GO TO 2230
2280 IF H2(3)=0 THEN 2350
2290 PRINT "  Row ";H2(3);": _____HHHHHHHHHHHHHHH";
2300 INPUT Q$
2310 IF LEN(Q$)<=15 AND LEN(Q$)<>0 THEN 2350
2320 PRINT "          15 characters maximum!GG"
2330 GO TO 2290
2350 N1=0
2360 OPEN N$;1,"F",Y$
2365 DELETE 1510,2360
2370 DELETE D9
2375 M9=MEMORY
2380 DIM D9(L1,28)
2390 READ #1:Y$,I1,D9
2400 CLOSE 1
2410 N1=0
2420 N2=9999999
2430 Z2=Z1+13 MIN 27
2440 FOR I2=1 TO 3
2450     FOR I1=Z1+1 TO Z2+1
2460         IF H2(I2)=0 THEN 2530
2470         N3=D9(H2(I2),I1)
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2480          IF N3<=N1 THEN 2500
2490          N1=N3
2500          IF N3=>N2 THEN 2520
2510          N2=N3
2520          GO TO 2540
2530          I1=Z2+1
2540      NEXT I1
2550 NEXT I2
2560 IF N2=>0 THEN 2580
2570 N2=0
2580 PRINT "JJJThe minimum Y value of the selected Rows is ";INT(N2)
2590 PRINT "and the maximum Y value is ";INT(N1)
2600 PRINT "JIndicate graph's:"
2610 PRINT "  Minimum Y value: _____HHHHHH";
2620 INPUT H5
2630 PRINT "  Maximum Y value: _____HHHHHH";
2640 INPUT H6
2650 PRINT "  Interval       : _____HHHHHH";
2660 INPUT H7
2670 I3=(H6-H5)/H7
2680 I3=I3-INT(I3)
2690 IF I3=0 THEN 2720
2700 PRINT "Interval must evenly divide into the range!GG"
2710 GO TO 2580
2720 PRINT "JJJShall the Chart be generated on the Screen(32) or on"
2730 PRINT "the Plotter(1)? __HH";
2740 INPUT D1
2750 IF D1=32 OR D1=1 THEN 2780
2760 PRINT "                    type '32' or '1' please!GG"
2770 GO TO 2720
2780 IF D1=32 THEN 2840
2790 PRINT "JReady Plotter!GG"
2800 PRINT "To continue, type <CR>. ";
2810 INPUT Y$
2840 PRINT @32:"L"
2850 Z9=G5+G6*2+G7+G8
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
2860 MOVE @D1:G1,G4-Z9-L1*R1
2870 DRAW @D1:G3,G4-Z9-L1*R1
2880 DRAW @D1:G3,G4
2890 DRAW @D1:G1,G4
2900 DRAW @D1:G1,G4-Z9-L1*R1
2910 MOVE @D1:G1,G4-G5
2920 DRAW @D1:G3,G4-G5
2930 MOVE @D1:R4+G8,G4-G5-G6
2940 DRAW @D1:G3,G4-G5-G6
2950 MOVE @D1:R4+G8,G4-G5-G6*2
2960 DRAW @D1:G3,G4-G5-G6*2
2970 MOVE @D1:G1,G4-G5-G6*2-G7
2980 DRAW @D1:G3,G4-G5-G6*2-G7
2990 MOVE @D1:G1,G4-G5-G6*2-G7-G8
3000 DRAW @D1:G3,G4-G5-G6*2-G7-G8
3010 MOVE @D1:R4,G4-G5
3020 DRAW @D1:R4,G4-G5-G6*2-G7-G8-L1*R1
3030 MOVE @D1:R4+G8,G4-G5
3040 DRAW @D1:R4+G8,G4-G5-G6*2-G7-G8-L1*R1
3060 IF D1=32 THEN 3100
3070 PRINT @D1,25:90,90
3080 PRINT @D1,17:E5,E6
3090 GO TO 3120
3100 X6=E5
3110 Y6=E6
3120 FOR I1=10 TO 28 STEP 5
3130      V$=SEG(A$,A1(I1),2)
3140      IF V$="  " THEN 3410
3150      Z2=VAL(V$)
3160      U$=SEG(A$,A1(I1+1),2)
3170      Z3=VAL(U$)
3180      X$=SEG(A$,A1(I1+2),1)
3190      Y$=SEG(A$,A1(I1+3),30)
3210      IF X$="S" THEN 3340
3220      Z$=SEG(A$,A1(I1+4),30)
3240      MOVE @D1:5,G4-Z9-R1*(Z2-1)
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
3250      DRAW @D1:5,G4-Z9-R1*Z3
3270      IF D1=32 THEN 3430
3280      MOVE @D1:2,G4-Z9-R1*Z3+0.5
3290      PRINT @D1:Y$
3300      MOVE @D1:4.5,G4-Z9-R1*Z3+0.5
3310      PRINT @D1:Z$
3320      GO TO 3430
3340      MOVE @D1:3,G4-Z9-R1*(Z2-1)
3350      DRAW @D1:3,G4-Z9-R1*Z3
3370      IF D1=32 THEN 3430
3380      MOVE @D1:2.5,G4-Z9-R1*Z3+0.5
3390      PRINT @D1:Y$
3400      GO TO 3430
3410      I1=28
3420      GO TO 3490
3430      FOR I3=Z2 TO Z3
3440          IF X$="S" THEN 3470
3450          A3(I3)=1
3460          GO TO 3480
3470          A2(I3)=1
3480      NEXT I3
3490 NEXT I1
3500 IF D1=32 THEN 3530
3510 PRINT @D1,25:0,0
3530 GO TO 3760
3540 FOR I1=10 TO 28 STEP 6
3550      X$=SEG(A$,A1(I1),2)
3560      Y$=SEG(A$,A1(I1+1),2)
3570      Z$=SEG(A$,A1(I1+2),1)
3580      Z2=VAL(X$)
3590      Z3=VAL(Y$)
3610      IF Z$="  " THEN 3760
3630      IF Z$="2" THEN 3710
3660      FOR I2=Z2 TO Z3
3670          A2(I2)=1
3680      NEXT I2
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
3690      GO TO 3740
3710      FOR I2=Z2 TO Z3
3720          A3(I2)=2
3730      NEXT I2
3740 NEXT I1
3760 I2=0
3770 FOR I1=30 TO 61 STEP 2
3780      X$=SEG(A$,A1(I1),1)
3800      Y$=SEG(A$,A1(I1+1),30)
3810      I2=I2+1
3815      IF I2>L1 THEN 4020
3820      IF A2(I2)=0 THEN 3850
3830      Z1=G1+3.5
3840      GO TO 3890
3850      IF A3(I2)=0 THEN 3880
3860      Z1=G1+5.5
3870      GO TO 3890
3880      Z1=G1+0.5
3890      IF D1<>32 THEN 3950
3900      X7=Z1
3910      Y7=G4-Z9-R1*I2+0.5
3920      K$=Y$
3930      GOSUB 8240
3940      GO TO 3980
3950      PRINT @D1,17:E5,E6
3960      MOVE @D1:Z1,G4-Z9-R1*I2+0.5
3970      PRINT @D1:Y$
3980      IF X$="N" OR X$<>"Y" THEN 4030
3990      MOVE @D1:G1,G4-Z9-R1*I2
4000      DRAW @D1:G3,G4-Z9-R1*I2
4010      GO TO 4030
4020      I1=61
4030 NEXT I1
4050 X$=SEG(A$,A1(1),15)
4060 Y$=SEG(A$,A1(2),15)
4070 IF D1=32 THEN 4100
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
4080 PRINT @D1,17:E1,E2
4090 GO TO 4120
4100 X6=E1
4110 Y6=E2
4120 IF Y$="                " THEN 4380
4130 IF D1=32 THEN 4230
4140 MOVE @D1:0.5,G4-2.5
4150 PRINT @D1:X$
4160 MOVE @D1:0.6,G4-2.4
4170 PRINT @D1:X$
4180 MOVE @D1:0.5,G4-4.7
4190 PRINT @D1:Y$
4200 MOVE @D1:0.6,G4-4.6
4210 PRINT @D1:Y$
4220 GO TO 4510
4230 Y7=G4-2.4
4240 X7=0.5
4250 K$=X$
4260 GOSUB 8240
4270 Y7=G4-2.3
4280 X7=0.6
4290 GOSUB 8240
4300 Y7=G4-4.6
4310 X7=0.5
4320 K$=Y$
4330 GOSUB 8240
4340 Y7=G4-4.5
4350 X7=0.6
4360 GOSUB 8240
4370 GO TO 4510
4380 IF D1=32 THEN 4450
4390 MOVE @D1:0.5,G4-4
4400 PRINT @D1,17:E3,E4
4410 PRINT @D1:X$
4420 MOVE @D1:0.6,G4-3.9
4430 PRINT @D1:X$
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
4440 GO TO 4510
4450 X7=0.5
4460 Y7=G4-4
4470 X6=E3
4480 Y6=E4
4490 K$=X$
4500 GOSUB 8240
4510 IF D1=32 THEN 4560
4520 PRINT @D1,17:E5,E6
4530 MOVE @D1:8,G4-13
4540 PRINT @D1:"A C T I V I T Y"
4550 GO TO 4620
4560 X7=8
4570 X6=E5
4580 Y6=E6
4590 Y7=G4-13
4600 K$="A C T I V I T Y"
4610 GOSUB 8240
4620 X$=SEG(A$,A1(4),16)
4630 IF D1=32 THEN 4670
4640 MOVE @D1:G3-17,G4-4
4650 PRINT @D1:X$
4660 GO TO 4710
4670 X7=G3-17
4680 Y7=G4-4
4690 K$=X$
4700 GOSUB 8240
4710 X$=SEG(A$,A1(3),45)
4720 GOSUB 7670
4730 IF D1=32 THEN 4780
4740 PRINT @D1,17:E3,E4
4750 MOVE @D1:25.5,G4-4
4760 PRINT @D1:X$
4770 GO TO 4840
4780 X6=E3
4790 Y6=E4
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
4800 X7=25.5
4810 Y7=G4-4
4820 K$=X$
4830 GOSUB 8240
4840 X$=SEG(A$,A1(6),30)
4850 Y$=SEG(A$,A1(7),30)
4860 IF D1=32 THEN 4930
4870 PRINT @D1,17:E9,E0
4880 MOVE @D1:0.5,0
4890 PRINT @D1:Y$
4900 MOVE @D1:0.5,1.5
4910 PRINT @D1:X$
4920 GO TO 5030
4930 X6=E9
4940 Y6=E0
4950 X7=0.5
4960 Y7=0
4970 K$=Y$
4980 GOSUB 8240
4990 Y7=1.5
5000 X7=0.5
5010 K$=X$
5020 GOSUB 8240
5030 X$=SEG(A$,A1(5),6)
5040 IF D1=32 THEN 5080
5050 MOVE @D1:G3-5.7,2.5
5060 PRINT @D1:X$
5070 GO TO 5140
5080 X7=G3-5.7
5090 Y7=2
5100 K$=X$
5110 GOSUB 8240
5140 MOVE @D1:R4+G8,4
5150 DRAW @D1:R4+G8,G4-Z9-L1*R1-G8
5160 MOVE @D1:G3,G4-Z9-L1*R1-G8
5170 DRAW @D1:G3,4
```

TITLE	ABSTRACT NO.
Gantt Chart	51/07-0102/0

```
5180 IF D1=32 THEN 5210
5190 PRINT @D1,17:E7,E8
5200 GO TO 5260
5210 X6=E7
5220 Y6=E8
5260 J4=0
5270 J5=0
5280 Z1=N5
5290 Z1=Z1+1
5300 Z2=Z2+1
5310 Z2=Z1+13 MIN 42
5320 Z3=(G3-R4-G8)/(Z2-Z1+1)
5330 Z5=R4+G8+0.5-Z3
5340 Z7=G4-Z9+0.5
5360 FOR I1=Z1 TO Z2
5370     J4=J4+1
5380     J5=J5+1
5390     Z8=Z7
5400     Z5=Z5+Z3
5420     FOR I2=1 TO L1
5430         Z8=Z8-R1
5450         IF D9(I2,I1)<0 THEN 5790
5460         Y$=STR(D9(I2,I1))
5470         IF Y$<>" 1.0E-4" THEN 5500
5480         Y$="  0/00"
5490         GO TO 5680
5500         Y$=REP("",1,1)
5510         I3=POS(Y$,".",1)
5520         IF I3=0 THEN 5580
5530         IF LEN(Y$)=>8 THEN 5640
5540         FOR I3=1 TO 8-LEN(Y$)
5550             Y$=" "&Y$
5560         NEXT I3
5570         GO TO 5640
5580         IF LEN(Y$)=>6 THEN 5640
5590         FOR I3=1 TO 6-LEN(Y$)
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
5600            Y$=" "&Y$
5610        NEXT I3
5620        GO TO 5680
5640        I3=POS(Y$,".",1)
5650        IF I3=0 THEN 5680
5660        Y$=REP("/",I3,1)
5670        Y$=REP("",LEN(Y$)-1,2)
5680        IF D1=32 THEN 5730
5690        PRINT @D1,17:E7,E8
5700        MOVE @D1:Z5,Z8
5710        PRINT @D1:Y$
5720        GO TO 5790
5730        X6=E7
5740        Y6=E8
5750        X7=Z5
5760        Y7=Z8
5770        K$=Y$
5780        GOSUB 8240
5790    NEXT I2
5810    IF D1<>32 THEN 5880
5820    X7=Z5+Z3-4.5
5830    Y7=G4-Z9+G8+0.5
5840    K$=STR(C2(2))
5850    K$=REP("",1,1)
5860    GOSUB 8240
5870    GO TO 5910
5880    MOVE @D1:Z5+Z3-4.5,G4-Z9+G8+0.5
5890    PRINT @D1:C2(2)
5910    D5=C2
5920    C5=1
5930    GOSUB 8400
5940    IF D5(1)=C2(1) THEN 6330
5950    IF D1<>32 THEN 6040
5960    X7=Z5-(J4*Z3/2-1.5)+Z3/2
5970    Y7=G4-Z9+G8+0.5+G7
5980    B$=SEG(F$,4*C2(1)-4,4)
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
5990      IF B$<>"   J" THEN 6010
6000      B$="DEC"
6010      K$=B$
6020      GOSUB 8240
6030      GO TO 6100
6040      MOVE @D1:Z5-(J4*Z3/2-1.5)+Z3/2,G4-Z9+G8+G7+0.5
6050      B$=SEG(F$,4*C2(1)-4,4)
6060      IF B$<>"   J" THEN 6080
6070      B$="DEC"
6080      PRINT @D1:B$
6090      IF Z5+Z3-0.7>128 THEN 6170
6100      J4=0
6120      MOVE @D1:Z5+Z3-0.5,G4-G5-G6
6130      DRAW @D1:Z5+Z3-0.5,G4-Z9-L1*R1
6140      MOVE @D1:Z5+Z3-0.5,G4-Z9-L1*R1-G8
6150      DRAW @D1:Z5+Z3-0.5,4
6170      IF D5(3)=C2(3) THEN 6330
6180      IF D1<>32 THEN 6260
6190      X7=Z5-J5*Z3/2-3+Z3
6200      Y7=G4-G5-G6+0.5
6210      K$=STR(D5(3))
6220      K$="19"&K$
6230      K$=REP("",3,1)
6240      GOSUB 8240
6250      GO TO 6290
6260      MOVE @D1:Z5-J5*Z3/2-3+Z3,G4-G5-G6+0.5
6270      PRINT @D1:"19";D5(3)
6280      IF Z5+Z3-0.5>128 THEN 6330
6290      J5=0
6310      MOVE @D1:Z5+Z3-0.5,G4-G5-G6
6320      DRAW @D1:Z5+Z3-0.5,G4-G5
6330 NEXT I1
6350 IF D5(1)<>C2(1) THEN 6510
6360 IF D1<>32 THEN 6450
6370 X7=Z5-(J4*Z3/2-1.5)+Z3/2
6380 Y7=G4-Z9+G8+G7+0.5
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
6390 B$=SEG(F$,4*C2(1),4)
6400 IF B$<>"   J" THEN 6490
6410 B$="DEC"
6420 K$=B$
6430 GOSUB 8240
6440 GO TO 6510
6450 MOVE @D1:Z5-(J4*Z3/2-1.5)+Z3/2,G4-Z9+G8+G7+0.5
6460 B$=SEG(F$,4*C2(1),4)
6470 IF B$<>"   J" THEN 6490
6480 B$="DEC"
6490 PRINT @D1:B$
6510 IF D5(3)<>C2(3) THEN 6650
6520 IF D1<>32 THEN 6600
6530 X7=Z5-J5*Z3/2-3+Z3
6540 Y7=G4-G5-G6+0.5
6550 K$=STR(D5(3))
6560 K$=REP("",1,1)
6570 K$="19"&K$
6580 GOSUB 8240
6590 GO TO 6650
6600 MOVE @D1:Z5-J5*Z3/2+Z3-3,G4-G5-G6+0.5
6610 PRINT @D1:"19";D5(3)
6650 Z5=R4+G8-0.5-Z3
6660 H3=4
6670 H4=G4-Z9-L1*R1-G8
6680 Z5=R4+G8
6690 H8=(H4-H3)/((H6-H5)/H7)
6700 H9=H5-H7
6730 FOR I1=H3 TO H4 STEP H8
6740     H9=H9+H7
6750     Y$=STR(H9)
6760     Y$=REP("",1,1)
6770     IF LEN(Y$)=6 THEN 6880
6780     FOR I3=1 TO 6-LEN(Y$)
6790         Y$=" "&Y$
6800     NEXT I3
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
6810      IF D1<>32 THEN 6870
6820      X7=R4+G8-6.6
6830      Y7=I1-1.25
6840      K$=Y$
6850      GOSUB 8240
6860      GO TO 6890
6870      MOVE @D1:R4+G8-6.6,I1-1.25
6880      PRINT @D1:Y$
6890      MOVE @D1:R4,I1
6900      DRAW @D1:G3,I1
6910 NEXT I1
6930 FOR I1=1 TO 3
6940      IF H2(I1)=0 THEN 7630
6970      IF Z1<>1 THEN 7020
6980      M1=-1
6990      M2=-1
7000      M3=Z5
7010      GO TO 7070
7020      M1=Z5
7030      M2=INT(D9(H2(I1),Z1-1))
7040      M2=H3+(M2-H5)/(H6-H5)*(H4-H3)
7050      M3=Z5
7070      FOR I2=Z1 TO Z2
7080          M3=M3+Z3
7090          M4=INT(D9(H2(I1),I2))
7100          M4=H3+(M4-H5)/(H6-H5)*(H4-H3)
7110          GOSUB 7840
7120          M1=M3
7130          M2=M4
7140      NEXT I2
7160      M9=G4-L1*R1-G8-5-Z9
7170      IF I1=3 THEN 7490
7180      IF I1=2 THEN 7340
7190      M1=0.5
7200      M2=M9+1
7210      M3=5.5
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
7220      M4=M9+1
7230      N6=1
7240      GOSUB 7840
7250      IF D1=32 THEN 7290
7260      MOVE @D1:6,M9
7270      PRINT @D1:O$
7280      GO TO 7630
7290      X7=6
7300      Y7=M9
7310      K$=O$
7320      GOSUB 8240
7330      GO TO 7630
7340      M1=0.5
7350      M2=M9-3+1
7360      M3=5.5
7370      M4=M9-3+1
7380      N6=1
7390      GOSUB 7840
7400      IF D1=32 THEN 7440
7410      MOVE @D1:6,M9-3
7420      PRINT @D1:P$
7430      GO TO 7630
7440      X7=6
7450      Y7=M9-3
7460      K$=P$
7470      GOSUB 8240
7480      GO TO 7630
7490      M1=0.5
7500      M2=M9-6+1
7510      M3=5.5
7520      M4=M9-6+1
7530      N6=1
7540      GOSUB 7840
7550      IF D1=32 THEN 7590
7560      MOVE @D1:6,M9-6
7570      PRINT @D1:Q$
```

TITLE

Gantt Chart

ABSTRACT NO:

51/07-0102/0

```
7580      GO TO 7630
7590      X7=6
7600      Y7=M9-6
7610      K$=Q$
7620      GOSUB 8240
7630 NEXT I1
7640 PRINT "GGGGG"
7645 INPUT Z$
7650 RETURN
7670 K2=LEN(X$)
7680 FOR K1=LEN(X$) TO 1 STEP -1
7690      Z$=SEG(X$,K1,1)
7700      IF Z$<>" " THEN 7730
7710      K2=K2-1
7720      GO TO 7740
7730      K1=1
7740 NEXT K1
7750 IF K2=LEN(X$) THEN 7820
7760 Z$=" "
7770 FOR K1=1 TO (LEN(X$)-K2)/2-1
7780      Z$=Z$&" "
7790 NEXT K1
7800 X$=REP(Z$,K2+1,(LEN(X$)-K2)/2)
7810 X$=REP(Z$,1,0)
7820 RETURN
7840 IF M1<0 OR M2<0 OR M3<0 OR M4<0 THEN 8220
7850 IF M2>H4 OR M4>H4 THEN 8220
7860 IF M2<H3 OR M4<H3 THEN 8220
7870 M2=INT(M2)
7880 M4=INT(M4)
7890 IF N6=1 THEN 7910
7900 IF D9(H2(I1),I2)<0 OR D9(H2(I1),I2-1)<0 THEN 8220
7910 MOVE @D1:M1,M2
7920 IF I1=1 THEN 8200
7930 S1=M3-M1
7940 S2=M4-M2
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
7950 S3=SQR(S1↑2+S2↑2)
7960 L=1
7970 IF ABS(S2)<L AND ABS(S1)<L THEN 8210
7980 U1=L*(S1/S3)
7990 U2=L*(S2/S3)
8000 U3=2*U1
8010 U4=2*U2
8020 GO TO I1 OF 8040,8090,8150
8030 GO TO 8200
8040 FOR I4=0 TO M3-M1-U1*2 MAX M4-M2-U2*2 STEP 2*(U1 MAX U2)
8050     RDRAW @D1:U1,U2
8060     RMOVE @D1:U1,U2
8070 NEXT I4
8080 GO TO 8200
8090 FOR I4=0 TO M3-M1-U1 MAX M4-M2-U2 STEP U1 MAX U2
8100     RDRAW @D1:0,0
8110     RMOVE @D1:U1,U2
8120 NEXT I4
8130 MOVE @D1:M3,M4
8140 GO TO 8210
8150 FOR I4=0 TO M3-M1-U1*3 MAX M4-M2-U2*3 STEP 3*(U1 MAX U2)
8160     RDRAW @D1:U3,U4
8170     RMOVE @D1:U1,U2
8180 NEXT I4
8190 DRAW @D1:M3,M4
8200 DRAW @D1:M3,M4
8210 N6=0
8220 RETURN
8240 FOR I=1 TO LEN(K$)
8250     J$=SEG(K$,I,1)
8260     J=POS(". B3AEFJCLGDHOOQTZ7S52XYMV968PRN]1I[KUW4",J$,1)
8270     IF J>0 THEN 8290
8280     J=1
8290     IF S(J+241)=0 THEN 8340
8300     MOVE X7+S(S(J+241))*X6*0.1333,Y7+S(S(J+241)+1)*Y6*0.087
8310     FOR J1=S(J+241)+2 TO S(J+281) STEP 2
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
8320          PRINT @32,20:S(J1)*X6*0.1333+X7,S(J1+1)*Y6*0.087+Y7
8330      NEXT J1
8340      X7=X7+X6
8350 NEXT I
8360 RETURN
8400 IF C5=0 THEN 8500
8410 C6=SGN(C5)
8420 T1=2
8430 GO TO T1 OF 8440,8440,8630,8630,8650
8440 C2(2)=C2(2)+C1(T1)*C5
8450 C0(2)=28
8460 IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 8480
8470 C0(2)=29
8480 IF C2(2)<1 OR C2(2)>C0(C2(1)) THEN 8510
8490 C0(2)=28
8500 RETURN
8510 IF C6+1 THEN 8600
8520 C2(1)=C2(1)+C6
8530 IF C2(1)=>1 AND C2(1)<=12 THEN 8580
8540 C2(3)=C2(3)+C6
8550 C2(1)=C2(1)-12*C6
8560 IF T1>2 THEN 8530
8570 GO TO 8590
8580 IF T1>2 THEN 8670
8590 IF C6+1 THEN 8450
8600 C2(2)=C2(2)-C6*C0(C2(1))
8610 IF C6+1 THEN 8520
8620 GO TO 8450
8630 C2(1)=C2(1)+C1(T1)*C5
8640 GO TO 8530
8650 C2(3)=C2(3)+C5
8660 GO TO 8490
8670 C2(2)=C2(2) MIN C0(C2(1))
8680 GO TO 8490
8720 C4=0.25*C2(3)=INT(0.25*C2(3)) AND C2(1)>2
8730 IF C2(1)=1 THEN 8770
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
8740 FOR B3=1 TO C2(1)-1
8750     C4=C4+C0(B3)
8760 NEXT B3
8770 C4=C4+C2(2)+365*C2(3)+INT(0.25*(C2(3)-1))
8780 C8=INT(C4/7)+1
8790 C7=C4-(C8-1)*7+1
8800 RETURN
8850 GO TO T1 OF 8860,8860,8970,8970,9000
8860 C2=C3
8870 GOSUB 3970
8880 C5=C4
8890 C6=C8
8900 C2=B2
8910 GOSUB 3970
8920 IF T1=2 THEN 8950
8930 C8=C5-C4
8940 RETURN
8950 C8=C6-C8
8960 RETURN
8970 C8=INT((C3(1)+2)/C1(T1))-INT((B2(1)+2)/C1(T1))
8980 C8=12/C1(T1)*(C3(3)-B2(3))+C8
8990 RETURN
9000 C8=C3(3)-B2(3)
9010 RETURN
9050 H1=0
9060 C7=POS(R$,"/",1)
9070 C8=POS(R$,"/",C7+1)
9080 C2(2)=1
9090 B$=SEG(R$,(C8 MAX C7)+1,LEN(R$)-(C8 MAX C7))
9100 B$=B$&V$
9110 C2(3)=VAL(B$)
9120 IF C2(3)<>N3 AND C2(3)=>1 AND C2(3)=INT(C2(3)) THEN 9150
9130 E1=118
9140 GO TO 9500
9150 IF C7<>0 THEN 9190
9160 IF T1<>5 THEN 9130
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
9170 C2(1)=1
9180 RETURN
9190 B$=SEG(R$,1,C7-1)
9200 IF B$<>"END" THEN 9250
9210 IF T1<>5 AND C8<>0 THEN 9130
9220 C2(1)=12
9230 C2(2)=31
9240 RETURN
9250 B$=B$&V$
9260 C2(1)=VAL(B$)
9270 IF C2(1)=N3 OR C2(1)>12 OR C2(1)<>INT(C2(1)) OR C2(1)<1 THEN 9130
9280 IF C8<>0 THEN 9340
9290 IF T1<3 OR T1=5 THEN 9130
9300 C2(1)=C1(T1)*(C2(1)-1)+1
9310 C2(1)=C2(1)+C9(1)-C1(T1)*INT(C9(1)/C1(T1)-0.2)-1
9320 IF C2(1)>12 THEN 9130
9330 RETURN
9340 C0(2)=28
9350 IF 0.25*C2(3)<>INT(0.25*C2(3)) THEN 9370
9360 C0(2)=29
9370 B$=SEG(R$,C7+1,C8-C7-1)
9380 IF B$<>"END" THEN 9440
9390 IF T1<3 OR T1>4 THEN 9130
9400 C2(1)=C1(T1)*C2(1)
9410 IF C2(1)>12 THEN 9130
9420 C2(2)=C0(C2(1))
9430 RETURN
9440 B$=B$&V$
9450 C2(2)=VAL(B$)
9460 IF C2(2)=N3 OR C2(2)<>INT(C2(2)) OR C2(2)<1 THEN 9130
9470 IF C2(2)>C0(C2(1)) THEN 9130
9480 C0(2)=28
9490 RETURN
9500 PRINT "Invalid Date!GG"
9510 RETURN
9520 FOR R1=1 TO 42
```

TITLE	ABSTRACT NO:
Gantt Chart	51/07-0102/0

```
9530      FOR R2=1 TO L1
9540          PRINT "ROW ";R2;"COLUMN ";R1;": ";
9550          INPUT D9(R2,R1)
9560      NEXT R2
9570 NEXT R1
```

Tape 19 F-8-9

TEKTRONIX

4051

APPLICATIONS LIBRARY PROGRAM

TITLE		ABSTRACT NUMBER
Capital Budget		51/00-0103/0
ORIGINAL DATE August 1979	REVISION DATE	MEMORY REQUIREMENT 32K
AUTHOR David Mills	Tektronix, Inc. Beaverton, OR	PERIPHERALS Optional - 4662 Plotter

ABSTRACT

Statements: 712

Files: 1 ASCII Program
1 ASCII Data (Project Directory)
Automatically marks data files

The program will build and maintain files of current capital budget projects. Up to 50 projects may be stored on one tape, with standard information as:

- Rank
- Project description
- Budget (by quarter)
- Category
- Originator
- Actual cost
- Cumulative capital

Up to 30 sub-items may contribute to a project including:

- Item description
- CCA number
- Cost
- Date
- Quarter

All information may be examined or altered to correct errors or changes in budget, etc.

Charts may be produced for one, or all projects, giving total project cost, cumulative cost, budget vs. actual expenditures, etc.

Cumulative and quarterly graphs of budget vs. actual cost for any project, or the total of all projects may be drawn.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

KEY	FUNCTION
1.	LIST MENU
2.	ADD ITEM TO PROJECT
3.	ADD PROJECT
4.	CHANGE PROJECT PARAMETERS
5.	CHANGE ITEM PARAMETERS
6.	CHART FOR ONE PROJECT
7.	CHART OF ALL PROJECTS
8.	GRAPHICS
12.	DELETE ITEM FROM PROJECT
13.	DELETE ENTIRE PROJECT
20.	DELETE ALL PROJECTS

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

TAPE STRUCTURE

The tape file structure is maintained by the program; new files are automatically marked as additional projects are added.

File Number	Contents
1	ASCII Program
2	ASCII Data (all project titles, budgets, etc.)
3 on	ASCII Data (one project per file, up to 30 items each)

File #2, containing the project information, is always in memory, this allows quick access to each project file. As new projects are added, the new information is appended to File #2, and the last file is marked to hold the project items.

If a project is deleted, only the project file (2) must be altered, it's corresponding item file will be written over at the next addition of a project.

OPERATING INSTRUCTIONS

Press AUTOLOAD if the program is located on the first file. If not the first file:

```
FIND #  (file number)
OLD
RUN
```

Note that if the program is not located on the first file the program must be recoded.

The program automatically marks data files as they are needed.

The menu will appear, the User-Definable Keys or the actual key number may be pressed.

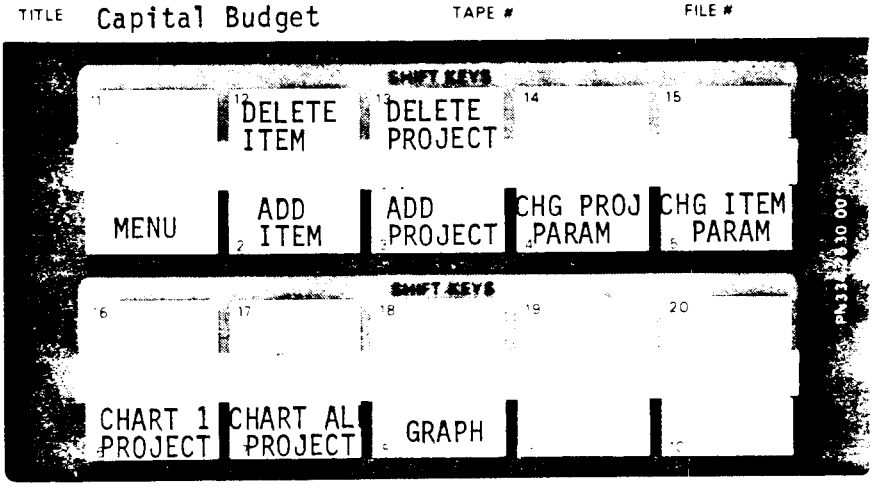
Projects and items may be added, or deleted in any order, though a project must exist before items can be added to it. If a project is an entity in itself, that, has no sub-items, its project name should also be entered as the only item. In this manner the project will serve as the budget, while the single item will represent total actual costs.

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

VARIABLE STRUCTURE

Variable	Used to Store	Type
B	Item cost	Array (N)
C1	Project category	Array (25*NØ)
C2	Project originating department	Array (NØ)
C3	Project budget	Array (NØ,4)
C9	Temporary variable	Simple
C$	Item date	String (8*N)
D$	Item description	String (25*N)
E	Temporary variable	Simple
E$	Item CCA number	String (F*N)
F	File number	Simple
FØ	File number	Simple
F$	Project descriptions	String (25*NØ)
I	Loop variable	Simple
I9	Loop variable	SImple
I$	Temporary variable	String (40)
J	Loop variable	Simple
J$	Temporary variable	String (10)
K	Temporary variable	Simple
K$	Temporary variable	String (10)
MØ	Temporary variable	Simple
M1	Window variable	Simple
M2	Window variable	Simple
N	Number of items per project	Simple
NØ	Number of projects	Simple
P	Project number	Simple
Q	Quarter in which item was initialized	Array (N)
S	Temporary variable	String (7)
T	Project summary	Array (5)
T1	Project summary	Simple

TITLE	ABSTRACT NUMBER
Capital Budget	51/00-0103/0



UDK #	Function
1	List menu
2	Add item to project
3	Add project
4	Change project parameters
5	Change item parameters
6	Chart for one project
7	Chart for all projects
8	Graphs
12	Delete item from project
13	Delete entire project

PLASTICS
CAPITAL BUDGET
YEAR-TO-DATE FINANCIAL STATEMENT

RANK	PROJECT IDENTIFICATION	CAT	ORIG	BUDGET	ACTUAL	CUM CAP
1	INSULATED HOPPERS & DRYER	1	4735	35	38	38
2	COMPARATOR (OPTICAL)	3	4726	18	15	53
3	CNC CONTROLLER	3	4733	140	20	73
4	MILLING MACHINES	3	4733	60	25	98
5	GRINDER	1	4733	45	45	143
6	LATHE	1	4747	25	25	168

* Cost stated in thousands of dollars

TITLE
Capital Budget

ABSTRACT NO:
51/00-0103/0

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

Variable	Used to Store	Type
X	Window variable	Array (4)
X1	Project category	Array (NØ)
X2	Quarter	Array (N)
X3	Item cost	Array (N)
X$	Temporary variable	String (25*N)
Y	Window variable	Array (4)
Y1	Window variable	Array (4)
Y$	Temporary variable	String (8*N)
Z	Output device	Simple
Z$	Temporary variable	String (7*N)

TITLE

Capital Budget

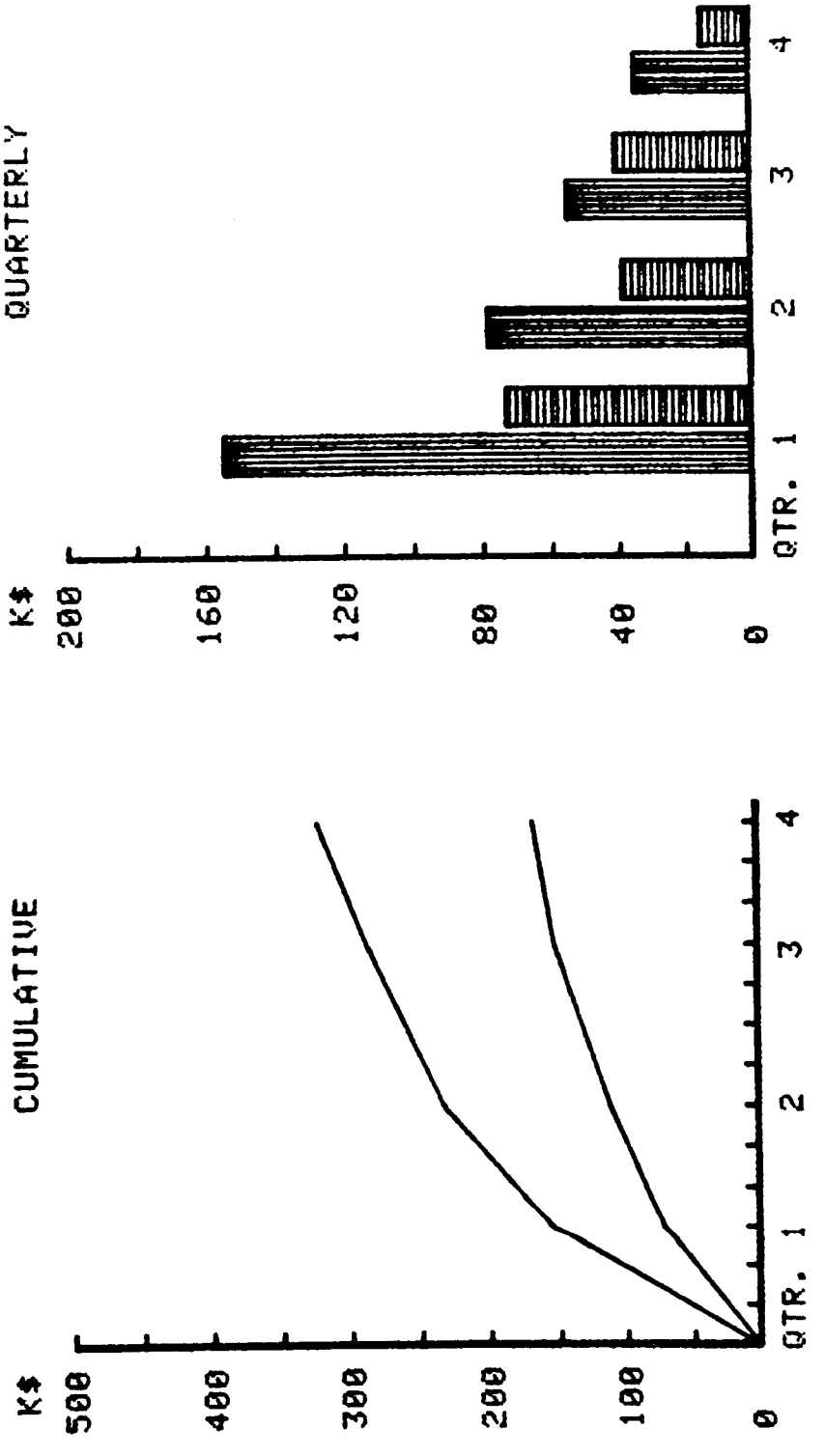
ABSTRACT NO:

51/00-0103/0

PLASTICS

Summary of capital plan status for FY 000

Year to date BUDGETED vs ACTUAL



PLASTICS
CAPITAL BUDGET
PROJECT FINANCIAL STATEMENT

PROJECT: INSULATED HOPPERS & DRYER
RANK: 1 CATEGORY: 1 ORIGINATOR: 4735

ITEM DESCRIPTION	CCA NUMBER	DATE	QTR	ITEM COST	CUM CAP
HOPPER #1	10X1003	06/12/79	1	10	10
DRYER	10X1007	08/08/79	1	8	18
HOPPER #2	10X1009	11/01/79	2	19	37
HOPPER REPAIR		12/10/79	3	1	38

PROJECT SUMMARY STATUS

QTR 1		QTR 2		QTR 3		QTR 4		TOTAL	
BUD	ACT	BUD	ACT	BUD	ACT	BUD	ACT	BUD	ACT
25	18	10	19	0	1	0	0	35	38

* Cost stated in thousands of dollars

TITLE
Capital Budget

ABSTRACT NO:
51/00-0103/0

TITLE

Capital Budget

ABSTRACT NO:

51/00-0103/0

```
1 GO TO 100

3 REM:      MENU

4 GOSUB 540
5 END

7 REM:      ADD ITEM TO PROJECT

8 GOSUB 1200
9 GO TO 4

11 REM:     ADD PROJECT

12 GOSUB 2290
13 GO TO 4

15 REM:     CHANGE PROJECT PARAMETERS

16 GOSUB 4060
17 GO TO 4

19 REM:   CHANGE ITEMS IN PROJECT

20 GOSUB 4520
21 GO TO 4

23 REM:     CHART FOR ONE PROJECT

24 GOSUB 2960
25 END

27 REM:     CHART FOR ALL PROJECTS

28 GOSUB 3710
29 END
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
31 REM:     GRAFIX

32 GOSUB 5170
33 END

47 REM:      DELETE ITEM FROM PROJECT

48 GOSUB 1960
49 GO TO 4

51 REM:      DELETE PROJECT

52 GOSUB 2750
53 GO TO 4

79 REM:  INITIALIZE ALL PROJECTS (CLEAR ALL DATA)

80 GOSUB 6750
81 GO TO 4
100 INIT

110 REM:
120 REM:      CAPITAL BUDGET
130 REM:
140 REM:      AUTHOR:  DAVID MILLS
150 REM:
160 REM:-----------------------------------------------------------
170 REM:
180 REM:              VARIABLE DEFINITIONS
190 REM:
200 REM:     N0     NUMBER OF PROJECTS
210 REM:     F$     STRING CONTAINING PROJECT DESCRIPTIONS
220 REM:     C1     PROJECT CATEGORY
230 REM:     C2     PROJECT ORIGINATING DEPT.
240 REM:     C3     PROJECT COST
```

TITLE	ABSTRACT NO:
Capital budget	51/00-0103/0

```
250 REM:     N      NUMBER OF ITEMS IN A PROJECT
260 REM:     D$     STRING CONTAINING ITEM DESCRIPTIONS
270 REM:     E$     ITEM CCA NUMBER
280 REM:     C$     ITEM DATE
290 REM:     Q      QUARTER IN WHICH EACH ITEM WAS INITIALIZED
300 REM:     B      ITEM COST
310 REM:

320 DIM J$(10),I$(40),K$(10),S(7)
330 DATA 1,1.5,2,2.5,5,7.5,10,32
340 READ S,Z
350 FIND 2
360 IF TYP(0)<>0 THEN 430
370 MARK 1,6144
380 FIND 2
390 N0=0
400 PRINT @33:N0
410 F$=""
420 GO TO 510
430 INPUT @33:N0
440 IF N0>0 THEN 470
450 F$=""
460 GO TO 510
470 DIM F$(25*N0),C1(N0),C2(N0),C3(N0,4)
480 F$=""
490 INPUT @33:F$
500 INPUT @33:C1,C2,C3
510 F=0
520 SET KEY

530 REM:
540 REM:      MENU
550 REM:      ----

560 PRINT "LJJIHKEY     FUNCTIONJ"
570 PRINT "I1.   LIST MENU"
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
580 PRINT "12.   ADD ITEM TO PROJECT"
590 PRINT "13.   ADD PROJECT"
600 PRINT "14.   CHANGE PROJECT PARAMETERS"
610 PRINT "15.   CHANGE ITEM PARAMETERS"
620 PRINT "16.   CHART FOR ONE PROJECT"
630 PRINT "17.   CHART OF ALL PROJECTS"
640 PRINT "18.   GRAPHICS"
650 PRINT "H12.   DELETE ITEM FROM PROJECT"
660 PRINT "H13.   DELETE ENTIRE PROJECT"
665 PRINT "H20.   DELETE ALL PROJECTS"
670 INPUT I
680 GO TO I OF 4,8,12,16,20,24,28,32,4,4,4,48,52,4,4,4,4,4,4,80
690 RETURN

700 REM:
710 REM:      LOAD FILE
720 REM:      ---- ----

730 IF F<3 THEN 830
740 DELETE D$,E$,C$,Q,B
750 FIND F
760 INPUT @33:N
770 IF N=0 THEN 830
780 DIM D$(25*N),E$(7*N),C$(8*N),Q(N),B(N)
790 INPUT @33:D$
800 INPUT @33:C$
810 INPUT @33:E$
820 INPUT @33:Q,B
830 RETURN

840 REM:
850 REM:      SAVE FILE
860 REM:      ---- ----

870 IF F<3 THEN 990
880 FIND F
```

TITLE

Capital Budget

ABSTRACT NO:

51/00-0103/0

```
890 IF TYP(0)<>0 THEN 910
900 MARK 1,2048
910 FIND F
920 PRINT @33:N
930 IF N=0 THEN 980
940 PRINT @33:D$
950 PRINT @33:C$
960 PRINT @33:E$
970 PRINT @33:Q,B
980 CLOSE
990 RETURN

1000 REM:
1010 REM:       SAVE PROJECT FILE
1020 REM:       ---- ------- ----

1030 FIND 2
1040 PRINT @33:N0
1050 IF N0=0 THEN 1080
1060 PRINT @33:F$
1070 PRINT @33:C1,C2,C3
1080 CLOSE
1090 RETURN

1100 REM:
1110 REM:       LIST AVAILABLE FILES
1120 REM:       ---- --------- -----

1130 PAGE
1140 FOR I=1 TO N0
1150      I$=SEG(F$,25*I-24,25)
1160      PRINT I;". ";I$
1170 NEXT I
1180 RETURN

1190 REM:
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
1200 REM:      ADD ITEM TO PROJECT
1210 REM:      --- ---- -- -------

1220 GOSUB 1110
1230 PRINT "JENTER PROJECT NUMBER ";
1240 F0=F
1250 INPUT F
1260 F=F+2
1270 IF F0=F THEN 1290
1280 GOSUB 710
1290 DELETE X$,Y$,Z$,X2,X3
1300 N=N+1
1310 DIM X$(25*N),Y$(8*N),Z$(7*N),X2(N),X3(N)
1320 X$=""
1330 Y$=""
1340 Z$=""
1350 IF N=1 THEN 1460
1360 FOR I=1 TO N-1
1370      I$=SEG(D$,25*I-24,25)
1380      X$=X$&I$
1390      J$=SEG(C$,8*I-7,8)
1400      Y$=Y$&J$
1410      K$=SEG(E$,7*I-6,7)
1420      Z$=Z$&K$
1430      X2(I)=Q(I)
1440      X3(I)=B(I)
1450 NEXT I
1460 PRINT "LJJENTER ITEM DESCRIPTION ";
1470 INPUT I$
1480 IF LEN(I$)=>25 THEN 1520
1490 FOR I=LEN(I$)+1 TO 25
1500      I$=I$&" "
1510 NEXT I
1520 I$=SEG(I$,1,25)
1530 X$=X$&I$
1540 PRINT "JENTER CCA NUMBER ";
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
1550 INPUT K$
1560 IF LEN(K$)=>7 THEN 1590
1570 K$=" "&K$
1580 GO TO 1560
1590 K$=SEG(K$,1,7)
1600 Z$=Z$&K$
1610 GOSUB 1780
1620 Y$=Y$&J$
1630 PRINT "ENTER QUARTER (1,2,3, or 4) ";
1640 INPUT X2(N)
1650 PRINT "ENTER ITEM COST ";
1660 INPUT X3(N)
1670 DELETE D$,C$,E$,Q,B
1680 DIM D$(25*N),C$(8*N),E$(7*N),Q(N),B(N)
1690 D$=X$
1700 C$=Y$
1710 E$=Z$
1720 Q=X2
1730 B=X3
1740 DELETE X$,Y$,Z$,X2,X3
1750 GOSUB 850
1760 RETURN

1770 REM:
1780 REM:      CORRECT DATE ENTRY
1790 REM:      ------- ---- -----

1800 PRINT "ENTER DATE (ie 10/31/79) ";
1810 INPUT J$
1820 IF LEN(J$)>8 THEN 1800
1830 I=POS(J$,"/",1)
1840 GO TO I OF 1800,1860,1870
1850 GO TO 1800
1860 J$="0"&J$
1870 I=POS(J$,"/",4)
1880 GO TO I-3 OF 1800,1900,1940
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
1890 GO TO 1800
1900 I$=SEG(J$,1,3)
1910 I$=I$&"0"
1920 J$=SEG(J$,4,4)
1930 J$=I$&J$
1940 RETURN

1950 REM:
1960 REM:      DELETE ITEM FROM PROJECT
1970 REM:      ------ ---- ---- -------

1980 GOSUB 1110
 990 PRINT "ENTER PROJECT NUMBER ";
2000 INPUT F
2010 F=F+2
2020 GOSUB 710
2030 PAGE
2040 FOR I=1 TO N
2050      I$=SEG(D$,25*I-24,25)
2060      PRINT I;". ";I$
2070 NEXT I
2080 PRINT "ENTER ITEM TO BE DELETED ";
2090 INPUT J$
2100 IF J$="" THEN 2270
2110 P=VAL(J$)
2120 IF P=N THEN 2230
 130 FOR I=P+1 TO N
2140      I$=SEG(D$,25*I-24,25)
2150      D$=REP(I$,25*(I-1)-24,25)
2160      I$=SEG(C$,8*I-7,8)
2170      C$=REP(I$,8*(I-1)-7,8)
2180      K$=SEG(E$,7*I-6,7)
2190      E$=REP(K$,7*(I-1)-6,7)
2200      Q(I-1)=Q(I)
2210      B(I-1)=B(I)
2220 NEXT I
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
2230 N=N-1
2240 IF N=0 THEN 2260
2250 DIM D$(25*N),C$(8*N),E$(7*N),Q(N),B(N)
2260 GOSUB 850
2270 RETURN

2280 REM:
2290 REM:      ADD PROJECT
2300 REM:      --- -------

2310 PRINT "[J]ENTER PROJECT IDENTIFICATION ";
2320 INPUT I$
2330 DELETE X$,X1,X2,X3
2340 N0=N0+1
2350 DIM X$(25*N0),X1(N0),X2(N0),X3(N0,4)
2360 IF N0=1 THEN 2440
2370 FOR I=1 TO N0-1
2380      X1(I)=C1(I)
2390      X2(I)=C2(I)
2400      FOR J=1 TO 4
2410          X3(I,J)=C3(I,J)
2420      NEXT J
2430 NEXT I
2440 IF LEN(I$)=>25 THEN 2480
2450 FOR I=LEN(I$)+1 TO 25
2460      I$=I$&" "
2470 NEXT I
 480 I$=SEG(I$,1,25)
2490 X$=F$&I$
2500 PRINT "[J]ENTER CATEGORY ";
2510 INPUT X1(N0)
2520 PRINT "[J]ENTER ORIGINATOR ";
2530 INPUT X2(N0)
2540 PRINT "[J]ENTER PROJECT BUDGET[J]"
2550 FOR I=1 TO 4
2560      PRINT "    QUARTER ";I;" ";
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
2570      INPUT X3(N0,I)
2580 NEXT I
2590 DELETE F$,C1,C2,C3
2600 DIM F$(25*N0),C1(N0),C2(N0),C3(N0,4)
2610 F$=X$
2620 C1=X1
2630 C2=X2
2640 C3=X3
2650 DELETE X$,X1,X2,X3
2660 GOSUB 1010
2670 FIND N0+2
2680 IF TYP(0)<>0 THEN 2700
2690 MARK 1,2048
2700 FIND N0+2
2710 PRINT @33:0
2720 CLOSE
2730 RETURN

2740 REM:
2750 REM:      DELETE PROJECT
2760 REM:      ------ -------

2770 GOSUB 1110
2780 PRINT "ENTER PROJECT TO BE DELETED ";
2790 INPUT P
2800 IF P<1 OR P>N0 THEN 2940
2810 IF P=N0 THEN 2880
2820 FOR I=P TO N0-1
2830      I$=SEG(F$,25*I-24,25)
2840      F$=REP(I$,25*(I-1)-24,25)
2850      C1(I)=C1(I+1)
2860      C2(I)=C2(I+1)
2870      C3(I)=C3(I+1)
2880      N0=N0-1
2890      IF N0>0 THEN 2930
2900      F$=""
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
2910      GO TO 2930
2920      DIM F$(25*N0),C1(N0),C2(N0),C3(N0)
2930      GOSUB 1010
2940      RETURN

2950      REM:
2960      REM:      CHART OF ONE PROJECT
2970      REM:      ----- -- --- -------

2980      F0=F
2990      GOSUB 1110
3000      PRINT "JENTER PROJECT NUMBER ";
3010      INPUT P
3020      IF P<1 OR P>N0 THEN 3090
3030      F=P+2
3040      IF F0=F THEN 3060
3050      GOSUB 710
3060      PAGE
3070      IF N>0 THEN 3100
3080      PRINT "JJTHERE ARE NO ITEMS IN THIS PROJECTGG"
3090      RETURN
3100      I$=SEG(F$,25*P-24,25)
3110      PRINT USING "32X,""PLASTICS""":
3120      PRINT USING "29X,""CAPITAL BUDGET""":
3130      PRINT USING "22X,""PROJECT FINANCIAL STATEMENT""":
3140      PRINT USING "72""-""":
3150      PRINT "| PROJECT: ";I$
3160      PRINT USING "71X,""K|""":
3170      PRINT "| RANK: ";P;"     CATEGORY: ";C1(P);"     ORIGINATOR: ";
3180      PRINT C2(P)
3190      PRINT USING "71X,""K|""":
3200      PRINT USING "72""-""":
3210      PRINT USING """K|"",70X,""|""":
3220      PRINT "|     ITEM DESCRIPTION    |CCA NUMBER|  DATE  |QTR|";
3230      PRINT "ITEM COST| CUM CAP  |"
3240      PRINT USING "72""-""":
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
3250      PRINT USING """K¦"",70X,""¦""":
3260      K=0
3270      IMAGE "¦",25A,"¦ ",8A," ¦",8A,"¦",1X,D,1X,"¦",6D,3X,"¦",6D,"    ¦"
3280      FOR I=1 TO N
3290          I$=SEG(D$,25*I-24,25)
3300          J$=SEG(C$,8*I-7,8)
3310          K$=SEG(E$,7*I-6,7)
3320          K=K+B(I)
3330          PRINT USING 3270:I$,K$,J$,Q(I),B(I),K
3340      NEXT I
3350      PRINT USING "72""-""":
3360      DELETE T
3370      DIM T(5)
3380      IF N<10 THEN 3430
3390      PRI USI """↑"",31""J"","""* Project summary on following page""":
3400      INPUT J$
3410      PAGE
3420      GO TO 3440
3430      PRINT USING """↑"",19""J""":
3440      PRINT USING "25X,""JJPROJECT SUMMARY STATUS""":
3450      PRINT USING "5X,61""-""":
3460      PRINT "      ¦   QTR 1   ¦   QTR 2   ¦   QTR 3   ¦   QTR 4   ¦";
3470      PRINT "   TOTAL   ¦"
3480      PRINT USING "5X, 61""-""":
3490      PRINT USING """K"",5X,""¦"",59X,""¦""":
3500      PRINT USING "5X,5""¦ BUD ¦ ACT "","""¦""":
3510      PRINT USING "5X,11""¦     """:
3520      PRINT USING """ KK"",5X,61""-""":
3530      PRINT USING """K"",5X,61"" """:
3540      PRINT USING "5X,11""¦     """:
3550      IMAGE "     ¦",6(4D," ¦")
3560      IMAGE "K",42X,4(4D," ¦")
3570      T=0
3580      T1=0
3590      FOR I=1 TO N
 600          T(Q(I))=T(Q(I))+B(I)
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
3610      NEXT I
3620      T(5)=SUM(T)
3630      T1=C3(F-2,1)+C3(F-2,2)+C3(F-2,3)+C3(F-2,4)
3640      PRINT USING 3550:C3(F-2,1),I(1),C3(F-2,2),T(2),C3(F-2,3),I(3)
3650      PRINT USING 3560:C3(F-2,4),T(4),T1,T(5)
3660      PRINT USING "5X,61""-""":
3670      PRI USI """T"",33""J"",""* Cost stated in thousands of dollars""":
3680      DELETE T
3690      RETURN

3700      REM:
3710      REM:      CHART WITH ALL PROJECTS
3720      REM:      ----- ---- --- --------

3730      PAGE
3740      IF N0>0 THEN 3770
3750      PRINT "JJCURRENTLY YOU HAVE NO PROJECTS ON FILE"
3760      RETURN
3770      PRINT USING "32X,""PLASTICS""":
3780      PRINT USING "29X,""CAPITAL BUDGET""":
3790      PRINT USING "20X,""YEAR-TO-DATE FINANCIAL STATEMENT""":
3800      PRINT USING "2X,68""-""":
3810      PRINT "  |RANK| PROJECT IDENTIFICATION   |CAT| ORIG |BUDGET";
3820      PRINT "|ACTUAL| CUM CAP |"
3830      PRINT USING "2X,68""-""":
3840      PRINT USING """K  |"",66X,""|""":
3850      K=0
3860      IMAGE"  |"3D" | ",25A,"|",2D," |",5D," |",5D," |",5D," |",7D,"  |"
3870      FOR I=1 TO N0
3880          C9=0
3890          F=I+2
3900          GOSUB 710
3910          I$=SEG(F$,25*I-24,25)
3920          IF N>0 THEN 3950
3930          I9=0
3940          GO TO 3970
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
3950          K=K+SUM(B)
3960          I9=SUM(B)
3970          FOR J=1 TO 4
3980              C9=C9+C3(I,J)
3990          NEXT J
4000          PRINT USING 3860:I,I$,C1(I),C2(I),C9,I9,K
4010      NEXT I
4020      PRINT USING "2X,68""-""":
4030      PRI USI """↑""33""J"","""  * Cost stated in thousands of dollars""":
4040      RETURN

4050      REM:
4060      REM:      CHANGE PROJECT VALUES
4070      REM:      ------ ------- ------

4080      GOSUB 1110
4090      PRINT "JJENTER PROJECT TO BE CHANGED: ";
4100      INPUT P
4110      I$=SEG(F$,25*P-24,25)
4120      PRINT "LJJJ1.  PROJECT NAME     : ";I$
4130      PRINT "2.  CATEGORY         : ";C1(P)
4140      PRINT "3.  ORIGINATOR       : ";C2(P)
4150      PRINT "4.  BUDGETJ"
4160      FOR I=1 TO 4
4170          PRINT "         QUARTER ";I;"    : ";C3(P,I)
4180      NEXT I
4190      PRINT "JJENTER THE PARAMETER NUMBER TO CHANGE ";
4200      INPUT J$
4210      IF J$="" THEN 4490
4220      GO TO VAL(J$) OF 4240,4330,4360,4390
4230      RETURN
4240      PRINT "JOLD PROJECT NAME WAS ";I$
4250      PRINT "ENTER NEW PROJECT NAME ";
4260      INPUT I$
4270      IF LEN(I$)=25 THEN 4310
4280      FOR I=LEN(I$)+1 TO 25
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
4290        I$=I$&" "
4300      NEXT I
4310      F$=REP(I$,25*P-24,25)
4320      GO TO 4460
4330      PRINT "JOLD CATEGORY WAS ";C1(P);" ENTER NEW CATEGORY ";
4340      INPUT C1(P)
4350      GO TO 4460
4360      PRINT "JOLD ORIGINATOR WAS ";C2(P);". ENTER NEW ORIGINATOR ";
4370      INPUT C2(P)
4380      GO TO 4460
4390      PRI "JENTER THE NEW BUDGET FOR THE PROJECT (RETURN FOR NO CHANGE)J"
4400      FOR I=1 TO 4
4410        PRI "  QUARTER ";I;". OLD VALUE WAS ";C3(P,I);" ENTER NEW VALUE
4420        INPUT J$
4430        IF J$="" THEN 4450
4440        C3(P,I)=VAL(J$)
4450      NEXT I
4460      PRINT "JJJANY MORE PARAMETERS TO CHANGE ? ";
4470      INPUT J$
4480      IF J$="Y" OR J$="YES" THEN 4120
4490      GOSUB 1010
4500      RETURN

4510      REM:
4520      REM:      CHANGE ITEMS IN PROJECT
4530      REM:      ------ ----- -- -------

4540      GOSUB 1100
4550      PRINT "JENTER PROJECT NUMBER TO CHANGE ";
4560      F0=F
4570      INPUT F
4580      F=F+2
4590      IF F0=F THEN 4630
4600      GOSUB 710
4610      IMAGE 3D,4X,25A,7A,2X,8A,5D,2X,7D
4620      I$=SEG(F$,25*(F-2)-24,25)
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
4630      PRINT "L                    PROJECT : ";I$;"J"
4640      PRINT "_NO.   ITEM DESCRIPTION        CCA_NUMBER    DATE    QTR";
4650      PRINT "   ITEM COST"
4660      PRINT USING "72""-""":
4670      FOR I=1 TO N
4680          I$=SEG(D$,25*I-24,25)
4690          J$=SEG(C$,8*I-7,8)
4700          K$=SEG(E$,7*I-6,7)
4710          PRINT USING 4610:I,I$,K$,J$,Q(I),B(I)
4720      NEXT I
4730      PRINT "JWHICH ITEM NEEDS TO BE CHANGED ? ";
4740      INPUT I9
4750      PRINT "L1.  ITEM DESCRIPTION : ";I$
4760      PRINT "2.  CCA NUMBER        : ";K$
4770      PRINT "3.  DATE              : ";J$
4780      PRINT "4.  QUARTER           : ";Q(I9)
4790      PRINT "5.  ITEM COST         : ";B(I9)
4800      PRINT "JENTER PARAMETER NUMBER TO CHANGE ";
4810      INPUT P
4820      GO TO P OF 4840,4930,5010,5060,5090
4830      RETURN
4840      PRINT "JOLD ITEM DESCRIPTION WAS ";I$
4850      PRINT "JENTER NEW DESCRIPTION ";
4860      INPUT I$
4870      IF LEN(I$)=25 THEN 4910
4880      FOR I=LEN(I$)+1 TO 25
4890          I$=I$&" "
4900      NEXT I
4910      D$=REP(I$,25*I9-24,25)
4920      GO TO 5110
4930      PRINT "JOLD CCA NUMBER WAS ";K$;" ENTER NEW NUMBER ";
4940      INPUT K$
4950      IF LEN(K$)=>7 THEN 4980
4960      K$=" "&K$
4970      GO TO 4950
4980      K$=SEG(K$,1,7)
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
4990      E$=REP(K$,7*I9-6,7)
5000      GO TO 5110
5010      PRINT "JOLD DATE WAS ";J$;" ENTER NEW DATE ";
5020      INPUT J$
5030      GOSUB 1820
5040      C$=REP(J$,8*I9-7,8)
5050      GO TO 5110
5060      PRINT "JOLD QUARTER WAS ";Q(I9);" ENTER NEW QUARTER ";
5070      INPUT Q(I9)
5080      GO TO 5110
5090      PRINT "JOLD ITEM COST WAS ";B(I9);" ENTER NEW COST ";
5100      INPUT B(I9)
5110      PRINT "JJANY MORE CHANGES ? ";
5120      INPUT J$
5130      IF J$="Y" OR J$="YES" THEN 4630
5140      GOSUB 850
5150      RETURN

5160      REM:
5170      REM:      GRAFIX
5180      REM:      ------

5190      DELETE X,Y,X1,Y1
5200      DIM X(4),Y(4),X1(4),Y1(4)
5210      X=0
5220      Y=0
5230      X1=0
5240      Y1=0
5250      GOSUB 1110
5260      PRINT "JJENTER PROJECT NUMBER, OR 'TOTAL' (T) ";
5270      INPUT J$
5280      IF J$="T" OR J$="TOTAL" THEN 5330
5290      P=VAL(J$)
5300      IF P<1 OR P>NO THEN 6730
5310      GOSUB 5380
5320      GO TO 5560
```

TITLE

Capital Budget

ABSTRACT NO:

51/00-0103/0

```
5330      FOR P=1 TO NO
5340          GOSUB 5380
5350      NEXT P
5360      GO TO 5560

5370      REM:
5380      REM:      SUB TO GET COSTS FOR PROJECT, OR TOTAL
5390      REM:      --- -- --- ----- --- -------- -- -----

5400      F=P+2
5410      GOSUB 710
5420      IF N=0 THEN 5550
5430      FOR I=1 TO 4
5440          X(I)=X(I)+C3(P,I)
5450          FOR J=1 TO N
5460              Y(I)=Y(I)+(Q(J)=I)*B(J)
5470          NEXT J
5480      NEXT I
5490      X1=X
5500      Y1=Y
5510      FOR I=2 TO 4
5520          X1(I)=X1(I)+X1(I-1)
5530          Y1(I)=Y1(I)+Y1(I-1)
5540      NEXT I
5550      RETURN
5560      PAGE
5570      MO=-9999
5580      FOR I=1 TO 4
5590          MO=MO MAX X(I) MAX Y(I)
5600      NEXT I
5610      GOSUB 5700
5620      M1=MO
5630      MO=-9999
5640      FOR I=1 TO 4
5650          MO=MO MAX X1(I) MAX Y1(I)
5660      NEXT I
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
5670      GOSUB 5700
5680      M2=M0
5690      GO TO 5800

 700      REM:      NEAT TIC ROUTINE

5710      FOR I=1 TO 10
5720          E=M0/10↑I
5730          IF E<10 THEN 5750
5740      NEXT I
5750      FOR J=1 TO 7
5760          IF E<=S(J) THEN 5780
5770      NEXT J
5780      M0=S(J)*10↑I
5790      RETURN
5800      FOR I=0 TO 1
5810          FOR J=0 TO 0.2 STEP 0.2
5820              PRINT @Z,21:15+(I=1)*65+J,70
 830              PRINT @Z,20:15+(I=1)*65+J,10+J*1.3
5840              PRINT @Z,20:60+(I=1)*65,10+J*1.3
5850          NEXT J
5860      NEXT I
5870      VIEWPORT 15,60,10,70
5880      WINDOW 0,13.5,0,5
5890      FOR I=1 TO 13
5900          MOVE @Z:I,0
5910          DRAW @Z:I,0.1
5920      NEXT I
5930      FOR I=0.5 TO 5 STEP 0.5
5940          MOVE @Z:0,I
5950          DRAW @Z:0.3,I
5960      NEXT I
5970      MOVE @Z:0,-1/3
5980      PRINT @Z:"QTR. 1      2      3      4"
5990      FOR I=0 TO 5
6000          J=M2/5*I
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
6010          MOVE @Z:-2.5,I-0.1
6020          PRINT @Z:J
6030       NEXT I
6040       WINDOW 0,13.5,0,M2
6050       MOVE @Z:0,0
6060       DRAW @Z:3,X1(1)
6070       DRAW @Z:6,X1(2)
6080       DRAW @Z:10,X1(3)
6090       DRAW @Z:13,X1(4)
6100       MOVE @Z:0,0
6110       FOR I=4 TO 2 STEP -1
6120          IF Y1(I)<>Y1(I-1) THEN 6140
6130       NEXT I
6140       IF I=1 AND Y1(1)=0 THEN 6190
6150       FOR J=1 TO I
6160          K=(J=1)*3+(J=2)*6+(J=3)*10+(J=4)*13
6170          DRAW @Z:K,Y1(J)
6180       NEXT J
6190       VIEWPORT 80,125,10,70
6200       WINDOW 0,13,0,5
6210       FOR I=0.5 TO 5 STEP 0.5
6220          MOVE @Z:0,I
6230          DRAW @Z:0.3,I
6240       NEXT I
6250       MOVE @Z:0,-1/3
6260       PRINT @Z:"QTR. 1     2     3     4"
6270       FOR I=0 TO 5
6280          J=M1/5*I
6290          MOVE @Z:-2.5,I-0.1
6300          PRINT @Z:J
6310       NEXT I
6320       WINDOW 0,13,0,M1
6330       FOR I=1 TO 4
6340          MOVE @Z:3*I-1,0
6350          RDRAW @Z:0,X(I)
6360          RDRAW @Z:0.9,0
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
6370          DRAW @Z:3*I-0.1,0
6380          FOR J=0.15 TO 0.75 STEP 0.15
6390              MOVE @Z:3*I-1+J,0
6400              RDRAW @Z:0,X(I)
6410          NEXT J
6420      NEXT I
6430      FOR I=1 TO 4
6440          IF Y(I)=0 THEN 6530
6450          MOVE @Z:I*3+0.1,0
6460          RDRAW @Z:0,Y(I)
6470          RDRAW @Z:0.9,0
6480          DRAW @Z:3*I+1,0
6490          FOR J=0 TO Y(I) STEP M1/100
6500              MOVE @Z:3*I+0.1,J
6510              RDRAW @Z:0.9,0
6520          NEXT J
6530      NEXT I
6540      PRINT @Z,21:65-4*1.79,95
6550      PRINT @Z:"PLASTICS"
6560      IF P>N THEN 6640
6570      I$=SEG(F$,25*P-24,25)
6580      FOR I=25 TO 1 STEP -1
6590          J$=SEG(I$,I,1)
6600          IF J$<>" " THEN 6620
6610      NEXT I
6620      PRINT @Z,21:(130-I*1.79)/2,90
6630      PRINT @Z:I$
 640      PRINT @Z,21:65-21*1.79,85
6650      PRINT @Z:"Summary of capital plan status for  FY 000"
6660      PRINT @Z,21:65-16*1.79,80
6670      PRINT @Z:"Year to date  BUDGETED vs ACTUAL"
6680      PRINT @Z,21:10,73
6690      PRINT @Z:"K$              CUMULATIVE"
6700      PRINT @Z,21:75,73
6710      PRINT @Z:"K$              QUARTERLY"
6720      DELETE X,Y,X1,Y1
```

TITLE	ABSTRACT NO:
Capital Budget	51/00-0103/0

```
6730      RETURN

6740      REM:
6750      REM:      INITIALIZE ALL PROJECTS (CLEAR ALL DATA)
6760      REM:      ---------- --- -------- ------ --- -----

6770      PRINT "LJJTHIS ROUTINE WILL DELETE ALL PROJECTS FROM THE TAPE."
6780      PRINT "DO YOU WISH TO CONTINUE?GGGGG ";
6790      INPUT I$
6800      IF I$<>"Y" AND I$<>"YES" THEN 6900
6810      FIND 2
6820      MARK 1,6000
6830      FIND 2
6840      PRINT @33:0,0
6850      CLOSE
6900      RETURN
```

TEKTRONIX® Tape 19 F-10

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
3D FUNCTION PLOT	51/00-5506/0
ORIGINAL DATE: January 1980 / REVISION DATE:	MEMORY REQUIREMENT: 8K
AUTHOR: Dony Robert, Brussels, Belgium	PERIPHERALS: Optional - 4662 Plotter

ABSTRACT

Statements: 206

Files: 1 ASCII Program

The program draws a 3D plot of a two-variable mathematical function (Z=F(X,Y), with the hidden lines removed from the plot. The user defines the function and enters real or representative data to obtain the plot. However, the program does not prompt for these inputs; the user simply enters them into the specified lines in the program.

The program provides the option of obtaining uniform scaling so that the plot surface fills the entire screen, which doesn't provide an accurate representation of the function, or obtaining true (rather than uniform) scaling. This option is set by changing a single program line.

The user may also set the following:

- interval on the X and Y axis
- azimuth angle in degrees
- dip angle in degrees
- number of lines on the surface
- number of points on each line
- file number to record the data on tape for later use or faster drawing

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

METHOD

The program draws a function of two variables Z = F(X,Y) with hidden line removal. The function Z is defined in a rectangular region so X1<=X<=X2 and Y1<=Y<=Y2.

PROGRAM LOADING

Press AUTOLOAD if the program is located on the first file on the tape otherwise:

```
FIND #  (program file number)
OLD
RUN
```

The program will draw the function plot on the screen.

PROGRAM OPERATION

An additional text example is given in lines 47 - 60.

Change the appropriate lines of code to create the plot desired.

line 110 - contains the read statement for line 120
READ X1,X2,Y1,Y2,O1,O2,L,N,F

line 120 - contains the data to be read in line 110
- X1,X2 is the interval on the X-axis
- Y1,Y2 is the interval on the Y-axis
- O1 is the azimuth angle in degrees
- O2 is the dip angle in degrees
- L is the number of lines on the surface
- N is the number of points on each line
- F is the file number to record the data on tape

line 370 - REM - will provide true scaling
GOTO 410 - will obtain a uniform scaling so that the surface fills the entire screen; but does not create a true representation of the function.

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

line 1230 - prints at the screen; change to PRINT @1,..... to output to the 4662 Plotter

line 1240 - same as line 1230

line 2000 - contains the function Z
for certain functions with vertical asymptotes or high-value for Z the functions MIN and MAX may be used to take the crest off the plot. Add the following lines:

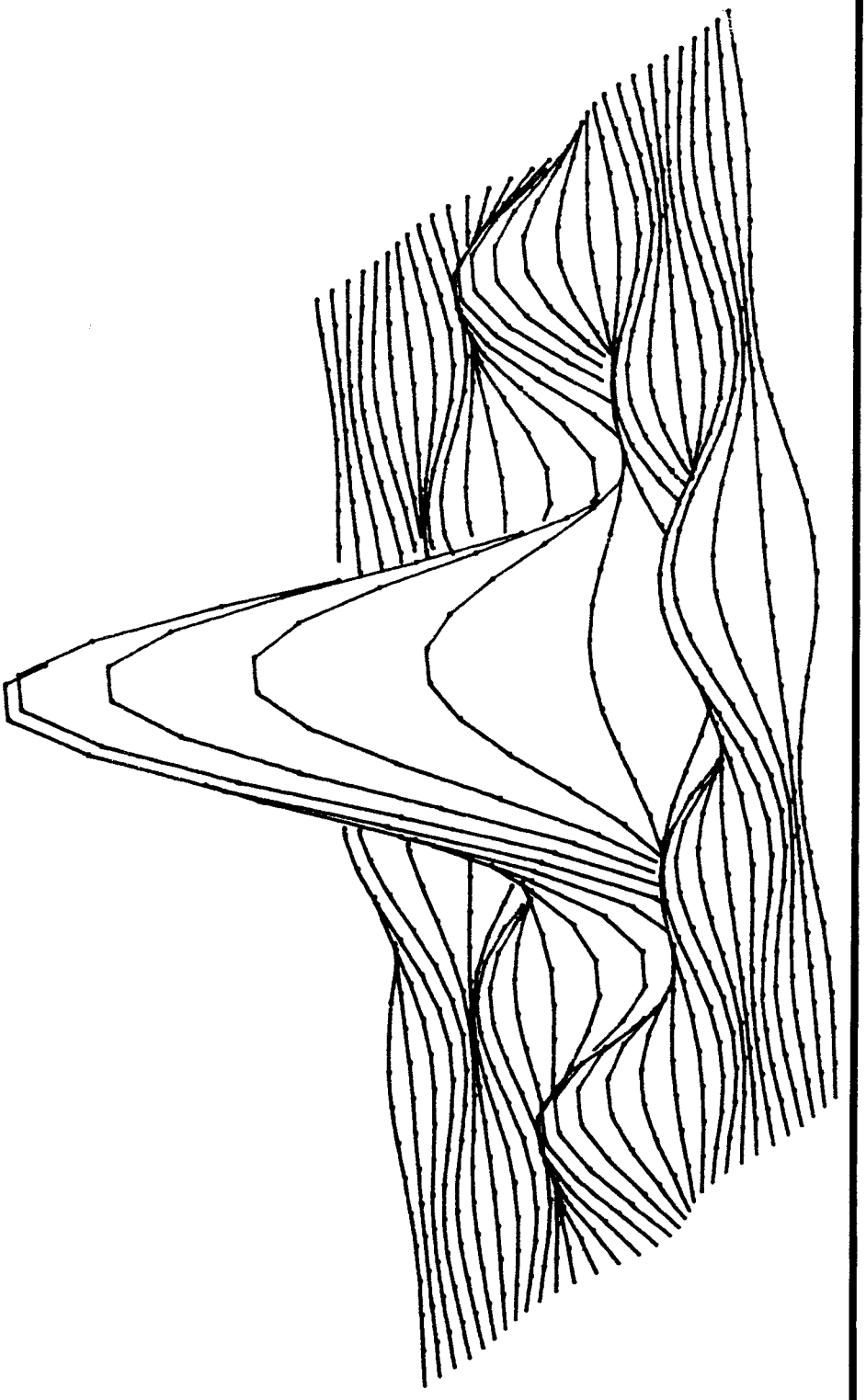
```
2001 Z=Z MIN 10
2002 Z=Z MAX -10
```

TITLE

3D FUNCTION PLOT

ABSTRACT NO:

51/00-5506/0



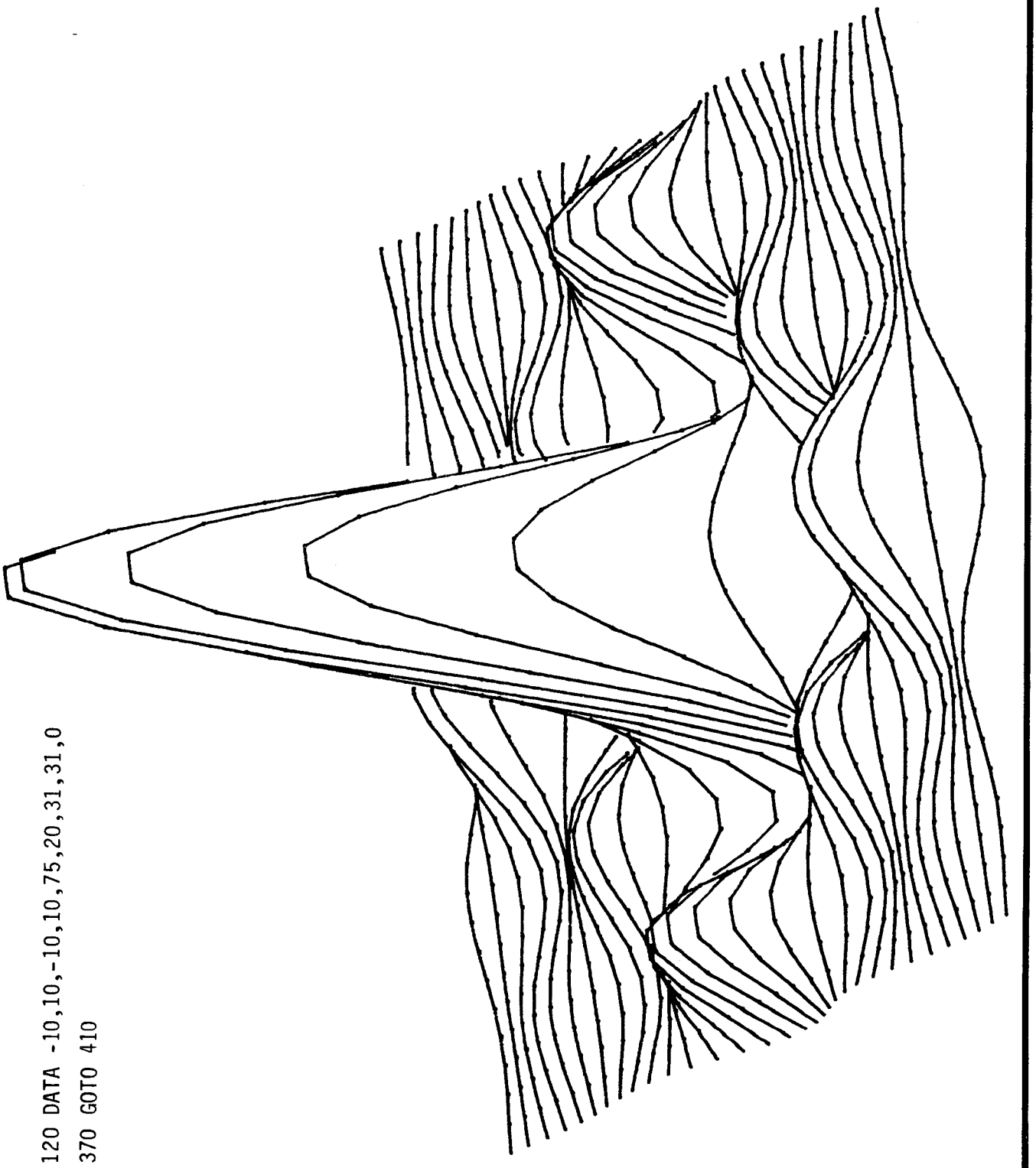
```
120 DATA -10,10,-10,10,75,20,31,31,0
370 REM
```

TITLE

3D FUNCTION PLOT

ABSTRACT NO:

51/00-5506/0



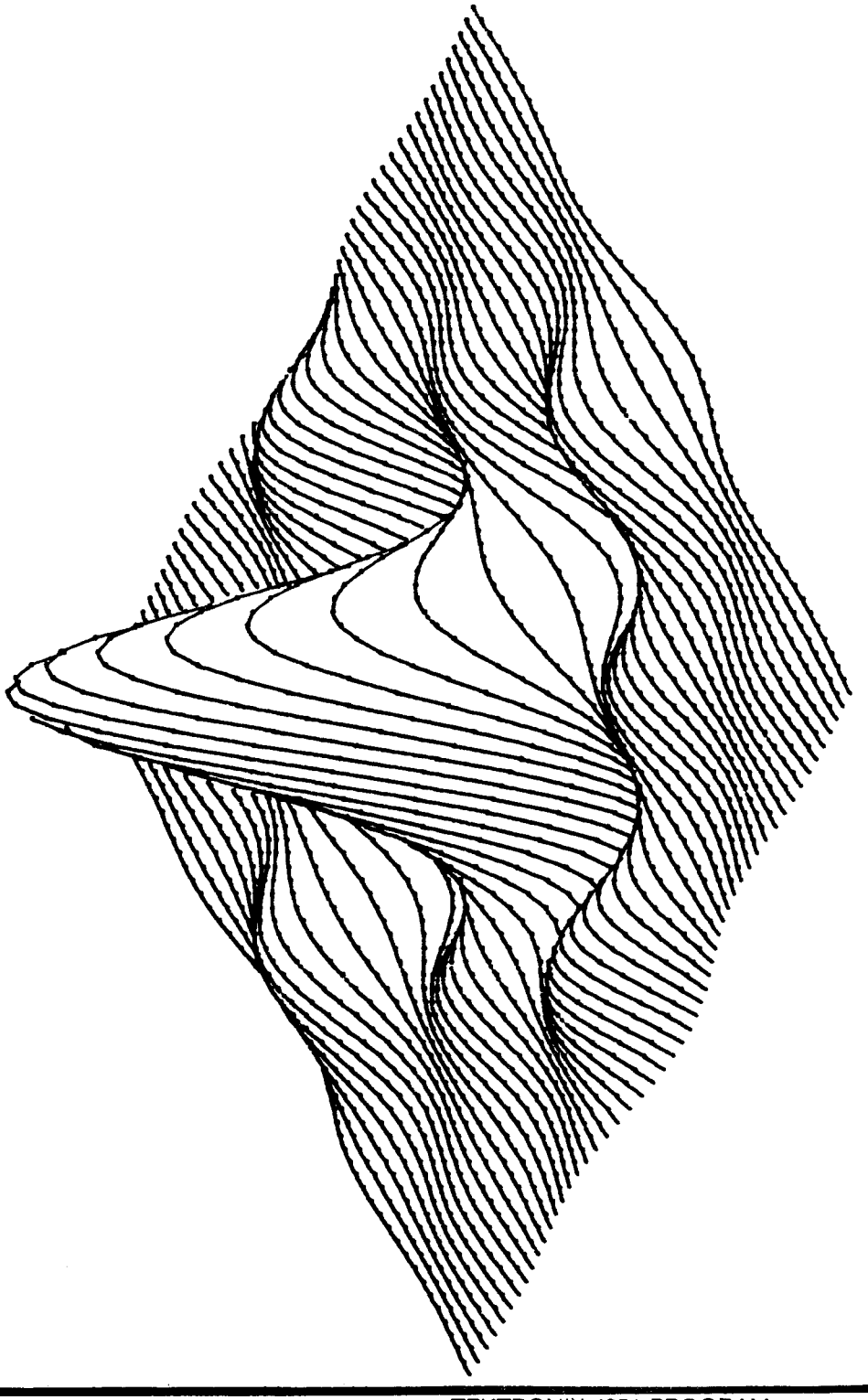
```
120 DATA -10,10,-10,10,75,20,31,31,0
370 GOTO 410
```

TITLE

3D FUNCTION PLOT

ABSTRACT NO:

51/00-5506/0



120 DATA -10,10,-10,10,45,30,51,51,0

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

INTERNAL DATA STRUCTURE

Variable	Used to Store	Type
A1	Function variable	Simple
A2	Function variable	Simple
A3	Temporary variable	Simple
A4	Temporary variable	Simple
B1	Temporary variable	Simple
B2	Temporary variable	Simple
B3	Temporary variable	Simple
B4	Temporary variable	Simple
C1	Cosine of O1	Simple
C2	Cosine of O2	Simple
D	Function variable	Simple
F	File number to record data on tape	Simple
G1	Temporary variable	Simple
H1	Scaling variable	Simple
H2	Scaling variable	Simple
I1	Temporary variable	Simple
I2	Temporary variable	Simple
I3	Temporary variable	Simple
I4	Temporary variable	Simple
K	Temporary variable	Simple
L	Number of lines on the surface	Simple
M	Temporary variable	Array
M1	Window size	Simple
M2	Window size	Simple
M3	Window size	Simple
M4	Window size	Simple
N	Number of points on each line	Simple
O1	Azimuth angle in degrees	Simple
O2	Dip angle in degrees	Simple

TITLE

3D FUNCTION PLOT

ABSTRACT NO:

51/00-5506/0

Variable	Used to Store	Type
P	Temporary variable	Simple
Q	Temporary variable	Simple
R	Function variable	Simple
S	Temporary variable	Simple
S1	Sine of 01	Simple
S2	Sine of 02	Simple
T	Temporary variable	Simple
U	Temporary variable	Simple
V	Temporary variable	Simple
X	Temporary variable	Simple
X0	Function variable	Simple
X1	Interval on X-axis	Simple
X2	Interval on X-axis	Simple
X3	Temporary variable	Simple
Y	Temporary variable	Simple
Y1	Interval on Y-axis	Simple
Y2	Interval on Y-axis	Simple
Y3	Temporary variable	Simple
Z	Plot function	Simple
Z1	Temporary variable	Simple
Z2	Temporary variable	Simple

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

```
1 REM..... 51/00-5506/0 ..... 3 D  F U N C T I O N  P L O T .....
2 REM
3 REM
4 REM.... THIS PROGRAM DRAWS A FUNCTION OF TWO VARIABLES Z=F(X,Y)
5 REM.... WITH HIDDEN LINES REMOVAL.
6 REM.... THE FUNCTION Z IS DEFINED IN A RECTANGULAR REGION  SO THAT
7 REM.... X1<=X<=X2  AND  Y1<=Y<=Y2.
8 REM.... TO OBTAIN A DRAW YOU MUST DO JUST LIKE THIS:
9 REM.... 1) WRITE THE FUNCTION Z IN LINE 2000
10 REM...    EXAMPLE:   2000 Z=10*SIN(X)/X*SIN(Y)/Y
11 REM... 2) WRITE THE DATA IN LINE 120 CORRESPONDING TO THE READ
12 REM...    STATEMENT IN LINE 110:
13 REM...       110 READ X1,X2,Y1,Y2,O1,O2,L,N,F    WHERE
14 REM
15 REM...  - [X1,X2] IS THE INTERVAL ON THE X-AXIS
16 REM...  - [Y1,Y2] IS THE INTERVAL ON THE Y-AXIS
17 REM...  - O1 IS THE AZIMUTH ANGLE IN DEGREES
18 REM...  - O2 IS THE DIP ANGLE IN DEGREES
19 REM...  - L IS THE NUMBER OF LINES ON THE SURFACE
20 REM...  - N IS THE NUMBER OF POINTS ON EACH LINE
21 REM...  - F IS THE FILE NUMBER TO RECORD THE DATA ON TAPE FOR
22 REM...    FASTEST DRAWING (NO RECORD IF F=0).
23 REM... 3) STATEMENT 370 MAY BE ... 370 GOTO 410  (1)  OR
24 REM...                             370 REM       (2)
25 REM...
26 REM...    IF (1),YOU OBTAIN A UNIFORM SCALING SO THAT THE SURFACE
27 REM...    FILLS THE ENTIRE SCREEN,THUS YOU DON'T OBTAIN THE TRUE
28 REM...    REPRESENTATION OF THE FUNCTION
29 REM...    IF (2),YOU OBTAIN THE TRUE SCALING
30 REM...
31 REM...
32 REM... 4) REMARKS:
33 REM       -------
34 REM       * ON THE 4051 IT TAKES ABOUT ONE MINUTE BEFORE DRAWING
35 REM       * ON THE 4052 AND 4054 DRAWING BEGINS ALMOST IMMEDIATELY
```

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

```
36 REM        * FOR CERTAIN FUNCTIONS WITH VERTICAL ASYMPTOTES OR
37 REM          HIGH-VALUE FOR Z,YOU CAN USE THE FUNCTIONS MIN
38 REM          AND MAX TO TAKE THE CREST OFF.
39 REM
40 REM          EXAMPLE:    2000 Z=...
41 REM                      2001 Z=Z MIN 10
42 REM                      2002 Z=Z MAX -10
43 REM...
44 REM...
45 REM... 5) TEST EXAMPLE
46 REM        ------------
47 REM        HERE IS A VERY NICE FUNCTION !
48 REM        YOU DO: -  2000 Z=10*SIN(X)/X*SIN(Y)/Y
49 REM                -  120 DATA -10,10,-10,10,45,30,51,51,0
50 REM                -  370 REM
60 REM                -  RUN

100 INIT
105 PAGE
110 READ X1,X2,Y1,Y2,O1,O2,L,N,F
120 DATA -10,10,-10,10,75,20,31,31,0
130 DIM M(2,130)

135 REM... INITIALIZATION FOR THE SUBROUTINE IN LINE 2000

140 SET DEGREES
150 S1=SIN(O1)
160 S2=SIN(O2)
170 C1=COS(O1)
180 C2=COS(O2)
190 SET RADIANS

192 REM... SEARCH FOR THE APROXIMATIVE MAXIMUM IN ABSOLUTE VALUE OF Z
193 REM... THE DIVIDER 19 IS ARBITRARY

200 M1=1000
```

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

```
210 M2=-1000
220 M3=M1
230 M4=M2
240 FOR Y=Y1 TO Y2 STEP (Y2-Y1)/19
250     FOR X=X1 TO X2 STEP (X2-X1)/19
260         GOSUB 2000
270         M1=M1 MIN XO
280         M2=M2 MAX XO
290         M3=M3 MIN YO
300         M4=M4 MAX YO
310     NEXT X
320 NEXT Y

325 REM... ENLARGEMENT TO BE SURE TO HAVE ALL THE PICTURE
326 REM... ON THE SCREEN

330 M1=M1-ABS(M1)/20
340 M2=M2+ABS(M2)/20
342 M3=M3-ABS(M3)/20
344 M4=M4+ABS(M4)/20

347 REM... SCALING...

350 H1=128/(M2-M1)
360 H2=98/(M4-M3)

370 REM... FOR TRUE SCALING...

380 IF H1<=H2 THEN 400
390 H1=H2
400 H2=H1

405 REM... F=NUMBER OF THE TAPE FILE TO RECORD THE DATA

410 IF F=0 THEN 440
420 FIND F
```

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

```
440 X3=(X2-X1)/N
450 Y3=(Y2-Y1)/L

460 REM... INITIALISATION OF THE MATRIX WHICH KEEP THE MAXIMAL AND
461 REM... MINIMAL CRESTS

490 M=-1
500 DIM M(2,65)
510 M=101
520 DIM M(2,130)

525 REM... START OF THE COMPUTATION OF THE LINES....

530 FOR U=1 TO L+1
540     IF O1>0 AND O1<180 THEN 546
542     Y=Y1+(U-1)*Y3
543     X=X2
544     G1=2
545     GO TO 590
546     Y=Y2-(U-1)*Y3
547     X=X1
548     G1=1
590     GOSUB 2000
600     A1=(X0-M1)*H1
610     B1=(Y0-M3)*H2
612     P=2
615     IF M(2,A1)=-1 THEN 670
620     P=0
630     IF M(2,A1)=>B1 THEN 650
640     P=2
650     IF M(1,A1)<=B1 THEN 670
660     P=1
670     FOR V=1 TO N
680         IF G1=2 THEN 700
690         X=X1+V*X3
695         GO TO 720
```

TITLE

3D FUNCTION PLOT

ABSTRACT NO:

51/00-5506/0

```
700         X=X2-V*X3
720         GOSUB 2000
730         A2=(X0-M1)*H1
740         B2=(Y0-M3)*H2
745         Q=P
750         IF M(2,A2)=-1 THEN 1140
780         S=0
790         Q=0
800         IF M(2,A2)=>B2 THEN 820
810         Q=2
820         IF M(1,A2)<=B2 THEN 840
830         Q=1
840         IF P=0 AND Q=0 THEN 1150
850         IF P=0 OR Q=0 THEN 950
860         IF P=Q THEN 1140
870         S=1
880         Z1=1
890         Z2=2
900         IF P=1 THEN 930
910         Z1=2
920         Z2=1
930         T=Z1
940         GO TO 960
950         T=P+Q
960         D=M(T,A2)-M(T,A1)-B2+B1
970         IF D=0 THEN 1140
980         A3=(B1*A2-B2*A1-M(T,A1)*A2+M(T,A2)*A1)/D
990         B3=(B1*M(T,A2)-B2*M(T,A1))/D
1000         IF P=0 THEN 1120
1010         A4=A2
1020         B4=B2
1030         A2=A3
1040         B2=B3
1050         GOSUB 1230
1060         A2=A4
1070         B2=B4
```

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

```
1080          IF S=0 THEN 1150
1090          T=Z2
1100          P=0
1110          GO TO 960
1120          A1=A3
1130          B1=B3
1140          GOSUB 1230
1150          P=Q
1160          A1=A2
1170          B1=B2
1180      NEXT V
1190 NEXT U
1210 HOME
1215 LIST 2000
1216 LIST 120
1220 END

1222 REM... SUBROUTINE WHICH DRAWS THE SURFACE,RECORDS THE DATA
1223 REM... ON THE TAPE AND UPDATE THE CRESTS MATRIX BY LINEAR
1224 REM... INTERPOLATION

1230 PRINT @32,21:A1,B1
1240 PRINT @32,20:A2,B2
1250 IF F=0 THEN 1270
1260 WRITE A1,B1,A2,B2
1270 I1=INT(A1+0.5)
1280 I2=INT(A2+0.5)
1290 IF I1<>I2 THEN 1330
1300 M(1,I2)=M(1,I2) MIN B2
1310 M(2,I2)=M(2,I2) MAX B2
1320 RETURN
1330 I3=(I1 MIN I2)+1
1340 I4=I1 MAX I2
1380 FOR K=I3 TO I4
1390      R=(K*(B2-B1)+A2*B1-A1*B2)/(A2-A1)
1400      M(1,K)=M(1,K) MIN R
```

TITLE	ABSTRACT NO:
3D FUNCTION PLOT	51/00-5506/0

```
1410      M(2,K)=M(2,K) MAX R
1420 NEXT K
1430 RETURN

1995 REM... SUBROUTINE WHICH CONTAINS THE FUNCTION Z AND CONVERTS IT
1996 REM... INTO THE (XO,YO) COORDINATE AFTER TWO ROTATIONS AND A
1997 REM... PROJECTION.

2000 Z=10*SIN(X)/X*SIN(Y)/Y
2010 XO=-X*S1+Y*C1
2020 YO=-X*C1*S2-Y*S1*S2+Z*C2
2030 RETURN
```

TEKTRONIX

4051

APPLICATIONS LIBRARY PROGRAM

TITLE		ABSTRACT NUMBER
RECORDKEEP II		51/07-6108/1
ORIGINAL DATE March 1979	REVISION DATE	MEMORY REQUIREMENT 16K
AUTHOR		PERIPHERALS 4907 File Manager Optional - 4631 HCU, 4641 Printer

ABSTRACT

Statements: 2188

Files: 14 ASCII Program

Tables and lists are two common ways in which people organize information to make comparisons or summarize data. RECORDKEEP II is a simple package that allows sorting such a table on the 4907 File Manager. Reports can then be quickly generated from this table (database) through the RECORD-KEEP II commands that allow editing, sorting, and table listing.

Twelve commands are available. They are broken down into three functional groups. The first group provides the commands necessary to edit the table, which we will call a database or file. Commands in the second group include file management operations that allow saving databases and creating smaller tables from existing databases. These tables can also be sorted and listed to emphasize their content by the third functional group. RECORDKEEP II can be applied in areas where you need to store and manipulate small inventories, sales history records, personnel records, patient history, or catalogues.

This software helps you organize and store information as well as put the information to work by summarizing it in a more effective tabular form.

Command 1 -- Add Items to Database
Command 2 -- Delete Items From Database
Command 3 -- Modify Items in Database
Command 4 -- List a Database
Command 5 -- Create a Local Database
Command 6 -- Archive a Local Database
Command 7 -- Make Local Databases From Archive
Command 8 -- List Directory
Command 9 -- Delete Archived Database
Command 10 -- Sort a Database
Command 11 -- List Database With Subtotals
Command 12 -- Stop

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

TABLE OF CONTENTS

PAGE

User Definable Key Description 2
RECORDKEEP II Overview 3

Section I - How To Start 3
Example 4

Section II - Loading RECORDKEEP Onto Disc 5
Example 5

Section III - Creating an Empty Database 6
Example 7

Section IV - Manipulate an Existing Database 9
Example - How to Begin 11

Command 1 - Add Items to Database 12
Example 13

Command 2 - Delete Items From Database 14
Example 14

Command 3 - Modify Items in Database 15
Example 15

Command 4 - List a Database 17
Example 18

Command 5 - Create a Local Database 19
Example 20

Command 6 - Archive a Local Database 22
Example 22

Command 7 - Make Local Databases From Archive 23
Example 23

Command 8 - List Directory 24
Example 24

Command 9 - Delete Archived Database 25
Example 25

Command 10 - Sort a Database 26
Example 26

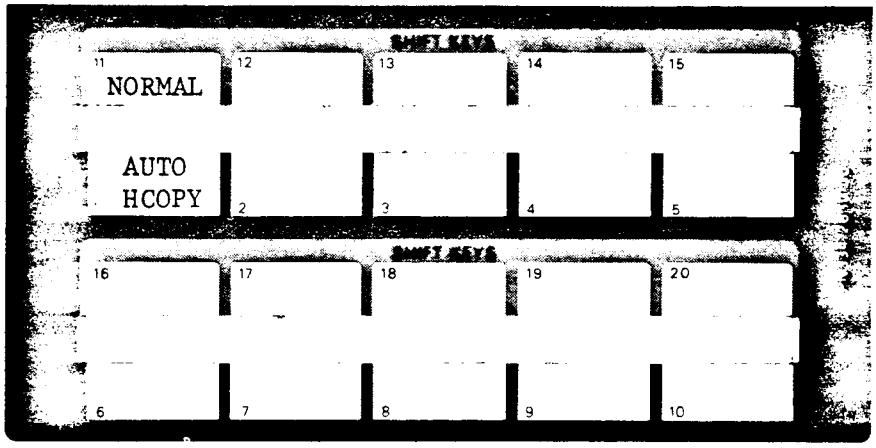
Command 11 - List Database With Subtotals 28
Example 29

Command 12 - Stop 31
Example 31

Disc File DATABASE/*/INIT/* Variable Descriptions . . . 32
Disc File DATABASE/*/DATA/* Variable Descriptions . . . 33
Tape Directory 33
Disc Library/File Structure for Software 34
Disc Library/File Structure for Databases 35
References 36

TITLE	ABSTRACT NUMBER
RECORDKEEP II	51/07-6108/1

TITLE RECORDKEEP II TAPE # FILE #



AUTO HCOPY - Automatically produces a hard copy on a 4631 when a "Page Full" condition occurs.

NORMAL - Forces the 4051 to pause and display a blinking "F" in the upper left hand corner of the screen when a "Page Full" condition occurs.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

OVERVIEW

This package actually consists of three different programs. One program creates a new database, another program adds information into this newly created database and allows the user to manipulate it, and the remaining program transfers the necessary programs from the 4051's internal tape drive onto the 4907. The user begins each session by inserting the program tape in the 4051 and pressing AUTO LOAD key. Before we go any further some general terms will be discussed as they apply to the following documentation.

RECORDKEEP II was designed to address an application where the user has a table of information that he wishes to store and from which he periodically retrieves all or part of the information and lists and/or sorts it to produce reports. The original table would be called a Master database. Master databases can be edited and listed, but can not be sorted. This database is always retained on disc, since it contains the most current information. Working files called Local databases can be created from a Master database or another Local file. These Local databases can be listed, sorted, and archived, but they can not be edited. Only the Master database can be edited. All Local databases are discarded (purged) when the program terminates normally. If the user wishes to save the information in a Local file for future use the file can be archived. Archived databases are not purged from the disc unless the user deliberately purges them. A Master database can have many Local and Archive databases associated with it, and all these databases are said to belong to the same database family. When the user creates a Master database, he assigns a family name to it that will apply to any other databases that are derived from it in the future.

Now back to describing the programs and how they work. The remainder of the documentation is divided into four sections, which are in the same order that the first time user would use them. The first section describes how to start RECORDKEEP II. The remaining three sections describe the 3 separate programs that make up RECORDKEEP II.

SECTION I
How To Start

The first step is to insert the program tape in the internal tape drive at the 4051 and then press the AUTO LOAD Key. The 4051 will give prompts for any informatin it needs and actions you must perform. All the necessary programs are stored on the tape cartridge. The program does not have the capability to format discs, but will prompt the user for all the information required to mount the disc(s). If the clock on the 4907 has not been set the program requests the date and time of the session in order to set it. The next questions ask for the device address where the program disc will be inserted, and the device where the databases will reside. After the device addresses have been entered the program asks the user to place each disc in their respective drives. When the program finishes mounting the disc(s) a menu will appear on the screen. Instructions for each program in this menu can be found in the following sections.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
                Data Base Management and Report Generation

Please enter the current date and time as
(DD-MMM-YY HH:MM:SS): 8-JAN-79 10:05:25

The disc with the programs will be placed in which device? 0

The disc with the database will be placed in which device? 0

Please insert the program/database disc in device 0.
Then press Return.
```

```
                Data Base Management and Report Generation

                              MENU
                  1)  Create a new master database?
                  2)  Manipulate an existing master database?
                  3)  Load system programs onto disc?

Please enter your selection:
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SECTION II

If the program disc does not contain the system programs, enter "3" as your selection from the menu in Section I. This will automatically transfer the programs from the tape cartridge to the program disc. This will take approximately 5 minutes and can be left unattended. Figure II shows what information appears on the screen when the transfer is complete. Once these programs have been placed on the program disc, the remaining two selections in the Section I menu can be run.

```
                Data Base Management and Report Generation

Please enter the current date and time as
(DD-MMM-YY HH:MM:SS): 8-JAN-79 10:05:25

The disc with the programs will be placed in which device? 0

The disc with the database will be placed in which device? 0

Please insert the program/database disc in device 0.
Then press Return.

                Data Base Management and Report Generation

                              MENU
                  1) Create a new master database?
                  2) Manipulate an existing master database?
                  3) Load system programs onto disc?

Please enter your selection: 3


*** FILEINIT TRANSFERED ***
*** MAIN TRANSFERED ***
*** OV1 TRANSFERED ***
*** OV2 TRANSFERED ***
*** OV3 TRANSFERED ***
*** OV3A TRANSFERED ***
*** OV4 TRANSFERED ***
*** OV5 TRANSFERED ***
*** OV6 TRANSFERED ***
*** OV7 TRANSFERED ***
*** OV7A TRANSFERED ***
*** OV8 TRANSFERED ***

***  Program transfer complete.  ***


Press AUTO LOAD key to select other programs.
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SECTION III

To create a new database first you must initialize it, which is selection 2 of the AUTOLOAD program. This asks questions about the general structure of the new database and what its contents will be, but before we begin let us define the terminology we will use. The database is a collection of numbers and text arranged in tabular form. Any given row in the table we will call an item and the columns of the table are called fields. The example below shows what the program will ask for.

It is best to spend some time to lay out what you want your Table (database) to look like. The program will ask you how many fields you want, the width of each field, the content of each field (text or numbers) and its titles. Planning in advance makes it easier to answer the questions asked and helps you catch some mistakes before they are made. The programs automatically inserts spaces between the fields to separate them so include at least one space between each field when you lay out your table. Armed with the table you constructed we can now quickly answer the questions and initialize the database.

The first question asked is how many additional fields will be in the database. The program automatically reserves the first field for the item numbers. It is 5 characters wide, a numeric field, and has a title of "ITEM#". This field will always be field 1. You need to enter the number of additional fields you will describe that make up the remainder of the database.

The next question asks for the width of the media the database will be displayed on. This should be the total width of the database, or put another way, the sum of the width of all fields plus the spaces separating each field can not be greater than the media width.

The program will then ask a series of questions for each field in your table. First it asks for the width of the field, then whether the content of the field will be numeric (numbers) or alphanumeric (text) and finally the field title, which can not be longer than the width of its field.

This information can be easily retrieved from the table you constructed on your scratch paper, and the program shows you the maximum length of the fields with underscore characters.

Once all the fields have been described, the program prompts you for a database family name. This name is used by other programs to select the correct database to manipulate in the future. You also have the option of making the database a private one. If you so choose the program asks for a password,

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1

which will limit any future use of the database to those who know the password. The database family name and the password each must be from 1 to 10 characters long, and start with a letter. The remaining characters have to be letters or digits (numbers). Do not put any blanks or other characters in these names.

These are the last questions asked and the program then writes the initilized database onto the disc and informs you that the initialization is complete. If you wish to add information to the database file you must select option 2 in the menu of the "AUTOLOAD" program described above.

Data Base Management and Report Generation

The disc with the programs will be placed in which device? 0

The disc with the database will be placed in which device? 0

Please insert the program/database disc in device 0.
Then press Return.

Data Base Management and Report Generation

MENU

1) Create a new master database?
2) Manipulate an existing master database?
3) Load system programs onto disc?

Please enter your selection: 1

FILE INITIALIZATION ROUTINE

This routine will ask for information regarding titles, size and the number of fields required for each item. The first field for each item will be a 5-character record number in the file and labeled as 'ITEM#'.

How many additional fields of information will be needed for each item: 3

Enter the width (in characters) of the media the data base will be displayed on. (e.g. for the 4051 screen it would be 72, for a printer it would be 132 characters)
The width is: 50

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
Enter, in order, width (number of characters) of each additional
field, whether the field will contain numeric (1) or alphanumeric (2)
data, and title for the field. The width of any field can not exceed
70 characters.

NOTE:The sum of all additional field widths must be
     less than or equal to 42.


User Field 2:  Field width (in characters):  5
Will entries be numeric (1) or alphanumeric (2):  1
Title for field
 :AGE__

User Field 3:  Field width (in characters):  5
Will entries be numeric (1) or alphanumeric (2):  1
Title for field
 :WI___

User Field 4:  Field width (in characters):  30
Will entries be numeric (1) or alphanumeric (2):  2
Title for field
 :PATIENTS_NAME_____________

Enter the database family name: PATIENTS__

Do you want this data base to be a private file? (Y/N): Y

Enter a file Password (1 to 10 characters must start with
 a leter but the remaining characters can be alphanumeric.)
Password is: RECORDS___

*
*** File initialization complete.
*


Press AUTO LOAD key to select other programs.
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SECTION IV

Selection "2" in the menu described in Section I is the most common selection. This program lets the user edit and manipulate existing databases. Before we begin the formal explanation of the 12 available commands some time should be spent on some general considerations that apply to many of the commands. If the program terminates abnormally, press the AUTOLOAD key and enter the start up information again. The program is segmented so simply typing RUN will not always give you the desired results. The program centers around a menu of 12 commands. Anytime the menu is present the user can select the appropriate command to accomplish his task.

There will be times when the user will enter the wrong menu number by mistake. The user can return to the menu and abort a command prematurely by either responding to a prompt for database name with a "!M" or typing zero (0) when asked for a field, or item number. At times you will be asked to name databases. These names must conform to the following rules They can be 1 to 10 characters long and can be composed of the characters A through Z, 0 through 9. The first character of the name must be a letter, and it can not have any leading or imbedded spaces. Even with these restrictions you have great flexibility in naming files. Below are some examples of good and bad database names:

Valid Names	Illegal Names	Reason Name Is Illegal
HENERY	ST RATE	imbedded blank
INVENTORY	ST.LOUIS	contains a decimal
RECORDS	INVENTORY	leading blank
A193	902ACTUALS	begins with number
B		

RECORDKEEP II maintains and manipulates existing databases using the 12 commands listed in the menu. It is assumed that the user has read the previous sections, since some of the terminology used is described there. The menu lists all the commands except for those under the control of user definable keys (UDK) #1 and #11. Key #1 can be used to get an automatic hard copy on a Tektronix 4631 when a page full condition occurs, while UDK #11 will revert the 4051 back to its normal mode of stopping on a page full condition and displaying a blinking "F" in the upper left hand corner of the screen. In this mode the screen must be manually paged (erased) by the user before the program will continue. Some caution should be used with UDK #1 if you do not have a hard copy unit (4631) or if the 4631 is turned off. The screen will immediately erase when the page full condition occurs and the user will not have time to read the information. There are a few instances when the 4051 will not respond

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

to pressing the UDK's. These are when a page full condition exists, and when the program is waiting for the user to enter in information. The best time to press the UDK's are when the 4051 is writing information on the screen.

The first prompt the program gives is to enter the database family name. This is the name used in Section III above to identify the database. If the database is private the password must immediately follow the family name separated a colon (e.g. FAMILY:PASSWORD). The next question asks where you wish the reports (menu selection 4 and 11) to be printed. Usually the screen is selected but if the database is wider than the screen (72 chars.) a printer (4641 character printer) would be more appropriate. Enter a zero (0) to select the screen, or enter a one (1) to choose the printer. If the printer is chosen the next request is for the primary (device) address of the printer.

After these questions have been answered the menu is displayed on the screen and you can select the command you wish by entering its number when asked. The 12 commands will be explained independently, but a reminder in case you find yourself in the wrong command by mistake. You can abort a command by entering "!M" for any prompt that asks for a database name or a zero (0) for any request for a field or item number.

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1

Data Base Management and Report Generation

MENU

1) Create a new master database?
2) Manipulate an existing master database?
3) Load system programs onto disc?

Please enter your selection: 2

RECORDKEEPING & SORTED REPORT GENERATOR

This program will maintain an inventory record and generate sorted reports at the user's request. At initial usage, the user must initialize the disc file through use of the file initialization routine.

Enter the desired database family: ACTUALS_____-

Do you want reports printed to screen(0) or printer (1): 0

MENU

1. Add items to master database file
2. Delete items from master database file
3. Examine or modify an item in master database file

4. List local or master database file
5. Create local database file
6. Archive local database file

7. Transfer archived database to local file
8. List directory of database files
9. Delete an archived database file

10. Sort local database file
11. List local or master database file with subtotals
12. Stop

Enter number desired:

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 1
Add items to master database file

The program will prompt you to enter data for each field. Each field prompt includes the field title and a series of underlined characters to indicate the field width. Items are always added to the end of the Master database (local and archived databases can never be added to). The program will scan all entries in numeric fields to verify they have valid numeric constants. Item numbers are automatically assigned by the program. To end the command you simply enter only a carriage return instead of data when the program begins another cycle of field prompts for the next item.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
                              MENU

                1.  Add items to master database file
                2.  Delete items from master database file
                3.  Examine or modify an item in master database file

                4.  List local or master database file
                5.  Create local database file
                6.  Archive local database file

                7.  Transfer archived database to local file
                8.  List directory of database files
                9.  Delete an archived database file

                10. Sort local database file
                11. List local or master database file with subtotals
                12. Stop


Enter number desired:  1


                        ROUTINE TO ADD ITEMS

Field titles will appear on the screen in order.  Enter data for
each item following the field title.  To discontinue data
entry, press RETURN after the first field title for an item appears.


Enter Ref #
 :Z38911
Enter Customer
 :UCI_________
Enter Product
 :4051___
Enter Qty
 :2__
Enter Dollars
 :16000__
Enter S.E.
 :SOBORI__
Enter A/P
 :Z04

Enter Ref #
 :______
```

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1

COMMAND 2
Delete items from master database file

This command begins by asking for the item number to be deleted. The item is then displayed on the screen and you are asked if you wish to delete the item. A "Y" to the question will delete the item and a "N" will prompt you for another item number. Each time an item is deleted you are asked if there are any more deletions. A "Y" will again prompt you for an item number where as a "N" will return to the menu. You can also return to the menu by entering a zero when asked for an item number. Before the program returns to the menu it automatically packs the database to remove all deleted items from the disc and this operation can be time consuming for large databases. Items can only be deleted from the Master database.

MENU

1. Add items to master database file
2. Delete items from master database file
3. Examine or modify an item in master database file

4. List local or master database file
5. Create local database file
6. Archive local database file

7. Transfer archived database to local file
8. List directory of database files
9. Delete an archived database file

10. Sort local database file
11. List local or master database file with subtotals
12. Stop

Enter number desired: 2

Enter Item No. to be deleted: 25

ITEM#	Ref #	Customer	Product	Qty	Dollars	S.E.	A/P
25	85C492	ROCKWELL	4631	1	4295	BEDORD	711

Is this the item you wish to delete (Y/N): Y

Do you wish to delete another item? (Y/N): N

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 3
Examine or modify an item in the MASTER database file.

This command begins by asking for the item number. This item is then displayed on the screen and you are asked if this is the item you wish to modify. If you respond with a "N" the prompt for an item number is repeated, or if you respond with a "Y" the program will begin prompting you for each field in the item, writting on the screen the field title, the current content (OLD:) of the field and a request for the new (NEW:) content. The underline characters are printed to indicate the maximum number of characters that you may enter. You may correct the field by typing in the correct contents followed by a return (CR) or you may leave contents alone by typing only a return (CR). After you have been prompted for all the fields in the item you are asked if you wish to modify another item. A "Y" response will produce a prompt for another item number while a "N" response will return you to the menu. Only the Master database may be modified with this command.

MENU

1. Add items to master database file
2. Delete items from master database file
3. Examine or modify an item in master database file

4. List local or master database file
5. Create local database file
6. Archive local database file

7. Transfer archived database to local file
8. List directory of database files
9. Delete an archived database file

10. Sort local database file
11. List local or master database file with subtotals
12. Stop

Enter number desired: 3

Enter Item No. to be examined: 15

ITEM#	Ref #	Customer	Product	Qty	Dollars	S.E.	A/P
15	85B954	HAC S	40141	1	12050	ETTER	713

Do you wish to modify this item (Y/N): Y

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
To leave a data item as it is, press RETURN.
To modify, enter correct data.

Ref #
OLD: 85B954
NEW
 :------

Customer
OLD: HAC  S
NEW
 :------------

Product
OLD: 40141
NEW
 :-------

Qty
OLD:   1
NEW
 :2__

Dollars
OLD:   12050
NEW
 :-------


S.E.
OLD: ETTER
NEW
 :--------

A/P
OLD: 713
NEW
 :---

Do you want to examine another item (Y/N):  N
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 4
List local or master database file

Two inputs are necessary for this command. The first input is the name of a local, or master database to be listed. To list a local database simply enter the name of the local database (presuming it exists). Master database can be selected even more simply by entering only a CR. The next input request is for a title that will be centered at the top of each new page. The maximum length of the title is printed and underline characters are also included to show the user the maximum length. Page numbers, and the current date and time (taken from the File Manager) are also printed on each page. The device the listings are sent to is determined by the answers to the questions asked when the program began. New pages are generated by issuing form feeds (ASCII character control L), and the program assumes that the printer can interpret these form feeds properly. When the listings are directed to the screen a "page full" condition signified by a blinking "F" in the upper right hand corner of the screen, will occur when the screen is filled. To continue with the listings the "PAGE" key must be pressed to erase the screen. User definable key #1 can be used to automatically produce a hard copy if a 4631 is available. This mode is activated by pressing user definable key #1 as the listing is placed on the screen. You may defeat this mode and return to the normal pause after each page full condition by pressing user definable key #11 while the listings are being written on the screen. When using the user definable keys there are two times when you should not press them. The first is when a blinking "?" is on the screen, indicating the 4051 is expecting a keyboard entry, and when a blinking "F" is seen in the upper left hand corner of the screen (page full condition).

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

MENU

1. Add items to master database file
2. Delete items from master database file
3. Examine or modify an item in master database file

4. List local or master database file
5. Create local database file
6. Archive local database file

7. Transfer archived database to local file
8. List directory of database files
9. Delete an archived database file

10. Sort local database file
11. List local or master database file with subtotals
12. Stop

Enter number desired: 4

Enter the name of the LOCAL database you want listed, or
enter only a CR if you want the MASTER database listed.
Database name? _____

Enter a 60 character title to be printed with listing.
:MASTER_DATABASE_LIST_TITLE_(THIS_IS_A_CENTERED_TITLE)_____

MASTER DATABASE LIST TITLE (THIS IS A CENTERED TITLE) PAGE 1
08-JAN-79 09:31

ITEM#	Ref #	Customer	Product	Qty	Dollars	S.E.	A/P
1	85A758	BERTEA	4662	1	3995	ADAMS	713
2	85A758	BERTEA	4662A01	1	175	ADAMS	713
3	85A849	UCI	4013	1	6850	ADAMS	713
4	85D066	CANNON	4012	1	6350	HENELY	713
5	85D066	CANNON	4953	1	3195	HENELY	713
6	85A653	UCI	4662	1	3995	ADAMS	712
7	85A630	FLUOR	4631	2	8590	ADAMS	712
8	85A661	AERO FORD	40101	1	4950	SABORI	712
9	85A661	AERO FORD	4662	1	3995	SABORI	712
10	85B723	CTY/ORANGE	4013	1	5495	ADAMS	712
11	85C697	SILICON SYS	40141	1	12910	ADAMS	712
12	85C710	WYLIE LABS	40141	1	12510	ADAMS	712
13	85A784	CAL POLY	4631	1	4295	WATSON	713
14	85A783	CAL POLY	4051	1	7500	WATSON	713
15	85B954	HAC S	40141	1	12050	ETTER	713
16	85B954	HAC	4631	1	4295	ETTER	713
17	85B994	IEC	4662	1	3995	BEDORD	713
18	85D097	GAI	4010	1	4450	BEDORD	713
19	85A740	CHEVRON	613	1	3195	BEDORD	712
20	85B860	OCC RES	40061	1	3320	BEDORD	712
21	85C686	ROCKWELL	4023	1	3495	BEDORD	712
22	64C499	AIR FORCE	RE4012	1	5826	ETTER	712
23	64C615	AIR FORCE	RE4012	1	5826	ETTER	712

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1

COMMAND 5

Create Local Database File

A Local database can be created from a source database (Master or Local database) via this command. The user specifies what characteristics the item in the source database must have in order to be included in the new local database. This is accomplished by a user inputted logical expression, which is applied to each item in the source database and if it is true the item is written in the new database. The remainder of this section assumes that the user understands how to write a BASIC logical expression. Information on how to write these expressions can be found in the 4051 BASIC Programmers Reference Manual, but the example near the end of the section may be sufficient to learn some simple expressions.

The first question asked is the database to be used as the source. If only a carriage return is entered from the keyboard the Master database will be selected, or the user may enter the name of a local database to be used as the source.

The next step is to choose fields in your database and assign them variable names so that they can be used in the logical expression. Numeric fields are first listed on the screen and user is prompted to enter the field number for the field you wish assigned to the variable name specified in the prompt. From zero to four numeric variable names can be assigned at one time, and to exit this portion of the command simply enter only a carriage return when asked for the field number. Immediately after the above portions is finished all the fields in your database will be listed. Out of this list from zero to four fields may be assigned to alphanumeric variable names in the same manner as the numeric variables were above. Next all the variable assignments are listed on the screen and the user is asked to enter a logical expression. The expression can contain only the variable names that have been assigned fields plus an numeric or alphanumeric constants (literals contained in quotes). Below are some examples of logical expressions (F1,A$, and B$ are assumed to be assigned to fields).

```
(F1<20 AND F1>100) AND A$="SCHEN" OR (B$<"L" AND B$>"C"
F1>0 AND F2<1000 AND A$ = B$ OR F3>F4
```

The expression entered can not be greater than 65 characters long, and must conform to the BASIC syntax rules for logical expressions. The final question asked is what name you wish to call the new local database. The program will then create the new local file, and examine each item in the source database. If the logical expression is true for any item it is written in the new database. When the menu returns to the screen the new database will contain all items in the source database that caused the logical expression to be true.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

MENU

1. Add items to master database file
2. Delete items from master database file
3. Examine or modify an item in master database file

4. List local or master database file
5. Create local database file
6. Archive local database file

7. Transfer archived database to local file
8. List directory of database files
9. Delete an archived database file

10. Sort local database file
11. List local or master database file with subtotals
12. Stop

Enter number desired: 5

Enter the name of the LOCAL database (i.e. Source database) from which you intend to create the new local database, or enter only a CR if you wish to use the MASTER database. _____

The next step is to assign database fields to variables which will be used in the following logical expression. This is done in two parts, and each can be terminated by responding with a CR to a prompt. Any numeric field maybe assigned to one of the variables F1,F2,F3, or F4, and similarly any field may be assigned to one of the character variables A$,B$,C$, or D$.

Numeric variables will be assigned first. Choose from list below.

1) ITEM#
5) Qty
6) Dollars
8) A/P

Enter the field number you wish to assign to variable F1: 6
Enter the field number you wish to assign to variable F2: 5
Enter the field number you wish to assign to variable F3:

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

Alphanumeric variables will now be assigned.

1) ITEM#
2) Ref #
3) Customer
4) Product
5) Qty
6) Dollars
7) S.E.
8) A/P

Enter the field number you wish to assign to variable A$: 7
Enter the field number you wish to assign to variable B$:

At this point you must enter a logical expression. When this expression is true for an item in the MASTER database it will be included in the LOCAL database being created. The only constraints on the expression are: 1) only data variables assigned to fields above can be used, 2) the expression must not have syntax errors, and 3) it can be no longer than 65 characters.
Fields are listed below.

F1 = Dollars
F2 = Qty

A$ = S.E.

Please enter the logical expression below.

IF A$="BEDORD"_AND_(F1*F2)>10000

Enter the name (1 to 10 characters) of the LOCAL database: PRODUCI

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 6

Archive Local Database File

Local files are deleted from the disc when command 12 is executed. A local file can be protected from deletion by archiving the database. This command needs only two inputs, the first is the name of the local file to be archived and the second input is the name you wish to call the archive database file. If the second name is currently in use, you are given the choice of choosing another name or overwriting the existing file and subsequently destroying the previous information in the file. After the archived file has been created the program returns to the menu.

```
                    MENU

          1.  Add items to master database file
          2.  Delete items from master database file
          3.  Examine or modify an item in master database file

          4.  List local or master database file
          5.  Create local database file
          6.  Archive local database file

          7.  Transfer archived database to local file
          8.  List directory of database files
          9.  Delete an archived database file

          10. Sort local database file
          11. List local or master database file with subtotals
          12. Stop

Enter number desired:  6

Enter the LOCAL database name you wish to archive: PRODUCI_

Enter the file name for the ARCHIVE copy, or if it is the same as
the local file name enter only a carriage return.
Enter name: BEDORD____
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 7

Transfer Archived Database to Local File

This is the compliment of Command 6. Again two inputs are needed. The name of the archived file you wish to transfer and the name you wish to give to the local database. You have the option of overwriting an existing local database if you wish. After the transfer is complete the menu is displayed.

```
                    MENU

          1.  Add items to master database file
          2.  Delete items from master database file
          3.  Examine or modify an item in master database file

          4.  List local or master database file
          5.  Create local database file
          6.  Archive local database file

          7.  Transfer archived database to local file
          8.  List directory of database files
          9.  Delete an archived database file

          10. Sort local database file
          11. List local or master database file with subtotals
          12. Stop

Enter number desired:  7

Enter the name of the ARCHIVED database to become a local file.
Enter name: BEDORD____

Enter the LOCAL file name, or only a carriage return if you want
the local file name the same as the archived file.
Enter name: STATUS____
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

Command 8

List Directory of Database Files

It is conceivable that one can become confused and forget what names he has assigned to the various databases. This command lists all database names for the database family selected at the very beginning of the program. No additional input is necessary and the page key must be pressed to return to the menu.

```
                    MENU

          1.  Add items to master database file
          2.  Delete items from master database file
          3.  Examine or modify an item in master database file

          4.  List local or master database file
          5.  Create local database file
          6.  Archive local database file

          7.  Transfer archived database to local file
          8.  List directory of database files
          9.  Delete an archived database file

          10. Sort local database file
          11. List local or master database file with subtotals
          12. Stop


Enter number desired:  8


Directory of database files
     08-JAN-79 09:46:14

   Master database
DATABASE/ACTUALS/MASTER/DATA

   Local databases
DATABASE/ACTUALS/LOCAL/DATA/PRODUCT
DATABASE/ACTUALS/LOCAL/DATA/STATUS
DATABASE/ACTUALS/LOCAL/DATA/REPORT

   Archive databases
DATABASE/ACTUALS/ARCHIVE/DATA/SORT
DATABASE/ACTUALS/ARCHIVE/DATA/BEDORD
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 9

Delete An Archived Database File

This command is the only means to purge archived database files. The only question asked is the name of the archived file you wish to destroy. Local file are automatically purged after command 12 is executed and the master database can never be purged with this program. The menu is displayed on the screen after the archived file has been purged.

MENU

1. Add items to master database file
2. Delete items from master database file
3. Examine or modify an item in master database file

4. List local or master database file
5. Create local database file
6. Archive local database file

7. Transfer archived database to local file
8. List directory of database files
9. Delete an archived database file

10. Sort local database file
11. List local or master database file with subtotals
12. Stop

Enter number desired: 9

Enter the name of the ARCHIVED file you want to delete from the disc.
Enter name: RECORD____

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 10

Sort Local Database File

Any local database can be sorted in place with this command. The user enters the name of the local file he wishes to sort and then the program displays a list of all fields in the database. The user can identify up to 4 fields to be used as keys in the sort. The user is asked to give the field number for the primary key and whether the key is sorted in ascending (A) or descending (D) order. In the same manner the user is asked to give field numbers and sorting order for the secondary, tertiary, and quartiary keys. If you don't need four keys, enter those keys you do want and initiate the actual sort by giving only a carriage reutrn when asked for a field number. The time it takes to sort the database depends mostly on the length and number of items in the database being sorted. A database with only 10 items may take less than 30 seconds while a database with 300 items may take as much as 30 minutes. After the sort is finished the program returns to the menu.

MENU

1. Add items to master database file
2. Delete items from master database file
3. Examine or modify an item in master database file

4. List local or master database file
5. Create local database file
6. Archive local database file

7. Transfer archived database to local file
8. List directory of database files
9. Delete an archived database file

10. Sort local database file
11. List local or master database file with subtotals
12. Stop

Enter number desired: 10

Enter the name of the LOCAL database you wish to sort.
Database name? REPORT____

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
You may choose up to four of the fields below to act as keys which
will control the resulting order of items in the sorted database.
The selection of keys is terminated by responding to a prompt with a CR.
      1) ITEM#
      2) Ref #
      3) Customer
      4) Product
      5) Qty
      6) Dollars
      7) S.E.
      8) A/P

Enter the field number to be used as the primary key: 3
Is the field to be sorted in ascending (A) or descending
 (D) order? A

Enter the field number to be used as the secondary key: 7
Is the field to be sorted in ascending (A) or descending
 (D) order? A

Enter the field number to be used as the tertiary key: 6
Is the field to be sorted in ascending (A) or descending
 (D) order? D

Enter the field number to be used as the quanternary key:
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 11

List Local or Master Database File With Subtotals

There are instances when one wishes to total a particular numeric field for all items where the content of a controlling field are equal. Suppose we have a database of accounts receivable that contain the account number and unit cost charged to the account. Now we sort the database by account number then list the database with subtotals for the dollars field and the control field being account number. This command would list each item in the database similar to the way command 4 does except when the controlling field changes (account number) a subtotal of the numeric field (dollars) is inserted in the list. When all items have been listed a grand total is also printed. The overall result of this example is a listing of the database with subtotals of costs charged to each account number and a grand total of all costs. Local and master databases may be listed with this command. When asked the user enters the name of the local database to be listed or only a carriage return if he wants the master database listed. The program then lists all numeric fields in the database and the user selects the one he wishes to be totaled. The program then lists all fields and the user selects a field that will be the control field. The next input is the title that will be printed at the top of each page. The listings look identical to the ones generated by command 4 except for the inclusion of subtotals and a grand total. The user definable keys #1 and #11 also operate as described under command 4.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
                              MENU

                1.  Add items to master database file
                2.  Delete items from master database file
                3.  Examine or modify an item in master database file

                4.  List local or master database file
                5.  Create local database file
                6.  Archive local database file

                7.  Transfer archived database to local file
                8.  List directory of database files
                9.  Delete an archived database file

                10. Sort local database file
                11. List local or master database file with subtotals
                12. Stop


Enter number desired:  11

Enter the name of the LOCAL database you want listed, or
enter only a CR if you want the MASTER database listed.
Database name? SORI______


Choose a numeric field to be totaled.

      1)  ITEM#
      5)  Qty
      6)  Dollars
      8)  A/P

Field number? 6


Choose a field below to control subtotals.

      1)  ITEM#
      2)  Ref #
      3)  Customer
      4)  Product
      5)  Qty
      6)  Dollars
      7)  S.E.
      8)  A/P

Field number? 8
```

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1

ACTUALS BILLED SUBTOTALED BY ACCOUNTING PERIOD PAGE 1
08-JAN-79 09:55

ITEM#	Ref #	Customer	Product	Qty	Dollars	S.E.	A/P
82	85H074	NAVY	4050A05	1	225	ADAMS	701
77	85H057	NAVY	4051	1	13045	BEDORD	701
78	85H057	NAVY	4631	1	3895	BEDORD	701
79	85H074	NAVY	4662	1	3995	CURE	701
80	85H074	NAVY	4924	1	2495	CURE	701
84	85H150	AIR FORCE	4631	1	3895	ETTER	701
83	85H074	NAVY	4050A06	1	400	ETTER	701
85	85H147	NAVY	40061	1	2995	HENELY	701
86	85H147	NAVY	4631	1	3995	HENELY	701
89	85H078	AIR FORCE	4050A03	1	400	SABORI	701
92	85H078	AIR FORCE	4050A06	1	400	WATSON	701
93	85H078	AIR FORCE	4051	1	11545	WATSON	701
			Subtotal		47285		
75	85J091	HAC	40141	1	13710	ADAMS	702
76	85J091	HAC	4631	1	4295	ADAMS	702
70	85H216	NAVY	40101	1	5070	CURE	702
71	85H268	ROCKWELL	4632	1	3995	CURE	702
72	85H293	ROCKWELL	4051	1	11600	ETTER	702
73	85H293	ROCKWELL	4631	1	4295	ETTER	702
74	85H293	ROCKWELL	4924	1	2495	ETTER	702
			Subtotal		45460		
69	85V215	HAC	4632	1	3995	BEDORD	703

ACTUALS BILLED SUBTOTALED BY ACCOUNTING PERIOD PAGE 6
08-JAN-79 09:55

ITEM#	Ref #	Customer	Product	Qty	Dollars	S.E.	A/P
4	85D066	CANNON	4812	1	6350	HENELY	713
5	85D066	CANNON	4953	1	3195	HENELY	713
98	81S929	MISSION RES	4051	1	2230	SABORI	713
100	81S929	MISSION RES	4631	1	859	SABORI	713
99	81S929	MISSION RES	4956	1	9800	SABORI	713
14	85A783	CAL POLY	4051	1	7500	WATSON	713
13	85A784	CAL POLY	4631	1	4295	WATSON	713
102	86F223	ROHR	4051	1	13550	WATSON	713
103	86F223	ROHR	4051R05	1	200	WATSON	713
			Subtotal		120879		
			Grand Total		568241		

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

COMMAND 12

Stop

This command terminates the program. Before all local files are deleted by this command the program asks if the user has archived all important local databases. If he responds with a "Y" then all local files are purged and execution stops. If the response is "N" then the program returns to the menu without purging any local databases.

```
                    MENU

          1.  Add items to master database file
          2.  Delete items from master database file
          3.  Examine or modify an item in master database file

          4.  List local or master database file
          5.  Create local database file
          6.  Archive local database file

          7.  Transfer archived database to local file
          8.  List directory of database files
          9.  Delete an archived database file

          10. Sort local database file
          11. List local or master database file with subtotals
          12. Stop


Enter number desired:  12

***  WARNING  -  LOCAL FILES WILL BE DELETED FROM DISC.
***  Have you archived those files you wish saved (Y/N)? Y

*
***  Execution complete.
*

Press AUTO LOAD key to select other programs.
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

Disc File DATABASE/*/INIT/* variable descriptions

INIT FILE — Generated with READ and WRITE

Record 1: T1, T2, T9, T8, T4, S1

Record 2: C, T$

Where:

T1 Number of fields per item.

T2 Number of items in the data base.

T9 Total of the field widths.

T8 Number of spaces automatically inserted between fields.

T4 The media width entered in FILEINIT (Section III)

S1 The total of field widths and the spaces inserted between the fields.

C An array that holds field characteristics (ITEM # is considered a field). The array is dimensioned C (T1,2). C(J,1) contains the width of the Jth field and C(J,2) holds the field type. "1" indicates the field is numeric and "2" indicates an alpha numeric field.

T$ A string variable containing all field titles with the appropriate number (T8) of spaces inserted between titles.

EXAMPLE:

T1 = 4; T2 = 0; T9 = 24; T8 - 2; T4 = 31; S1 = 30

```
C (4,2)  =  5  1      width of 5      numeric data
            8  2      width of 8      alphanumeric data
            6  1      width of 6      numeric data
            5  2      width of 5      numeric data


          123456789 123456789 123456789 123456789
T$ =      ITEM#  TITLE1    TITLE2  TITLE
          <------------ S1 ------------>
          <------------ T4 ------------>
```

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1

Disc File DATABASE/*/DATA/* Variable Descriptions

Accessed with READ and WRITE

This file is a collection of T2 character strings, each T4 characters long.

For the software to operate properly, the programs and overlays must reside on the first 13 files of a tape and must occur in the following order:

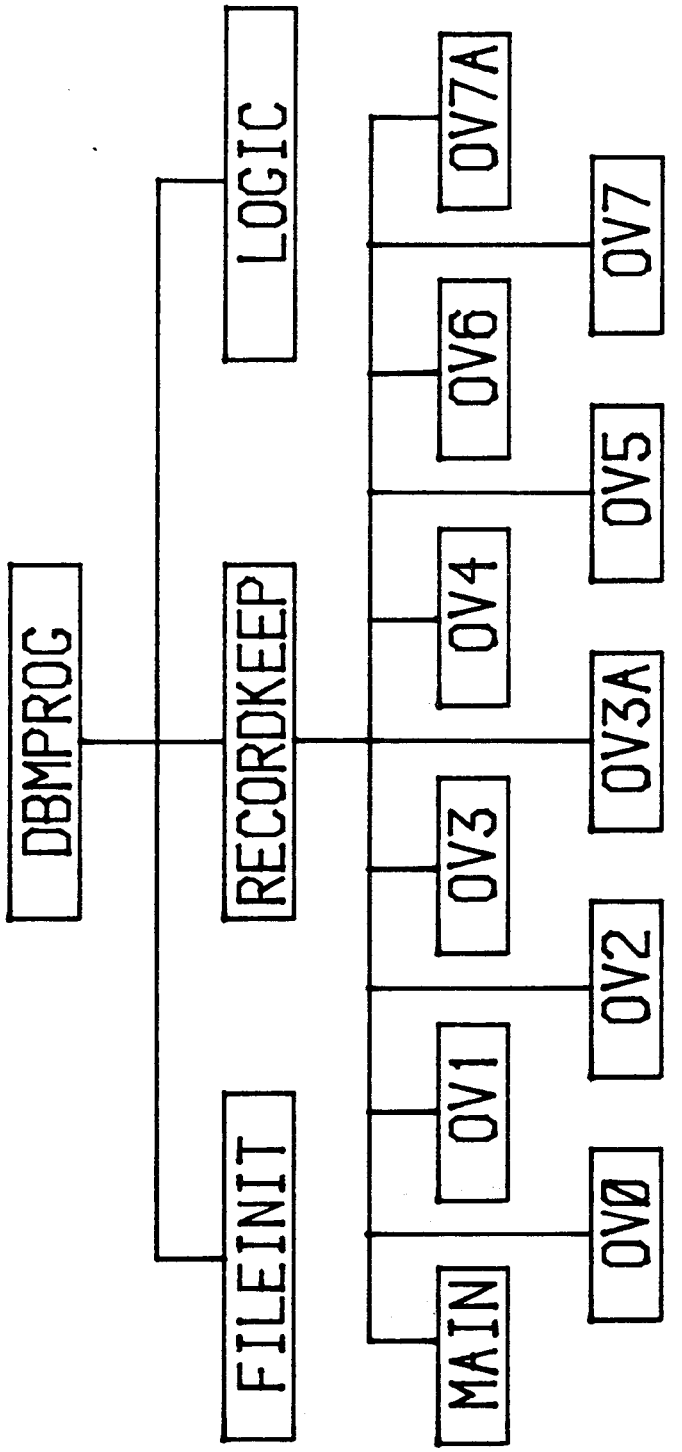
FILE	CONTENT
1	"Auto Load" program
2	FILEINIT
3	MAIN
4	OV1
5	OV2
6	OV3
7	OV3A
8	OV4
9	OV5
10	OV6
11	OV7
12	OV7A
13	OV8

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1



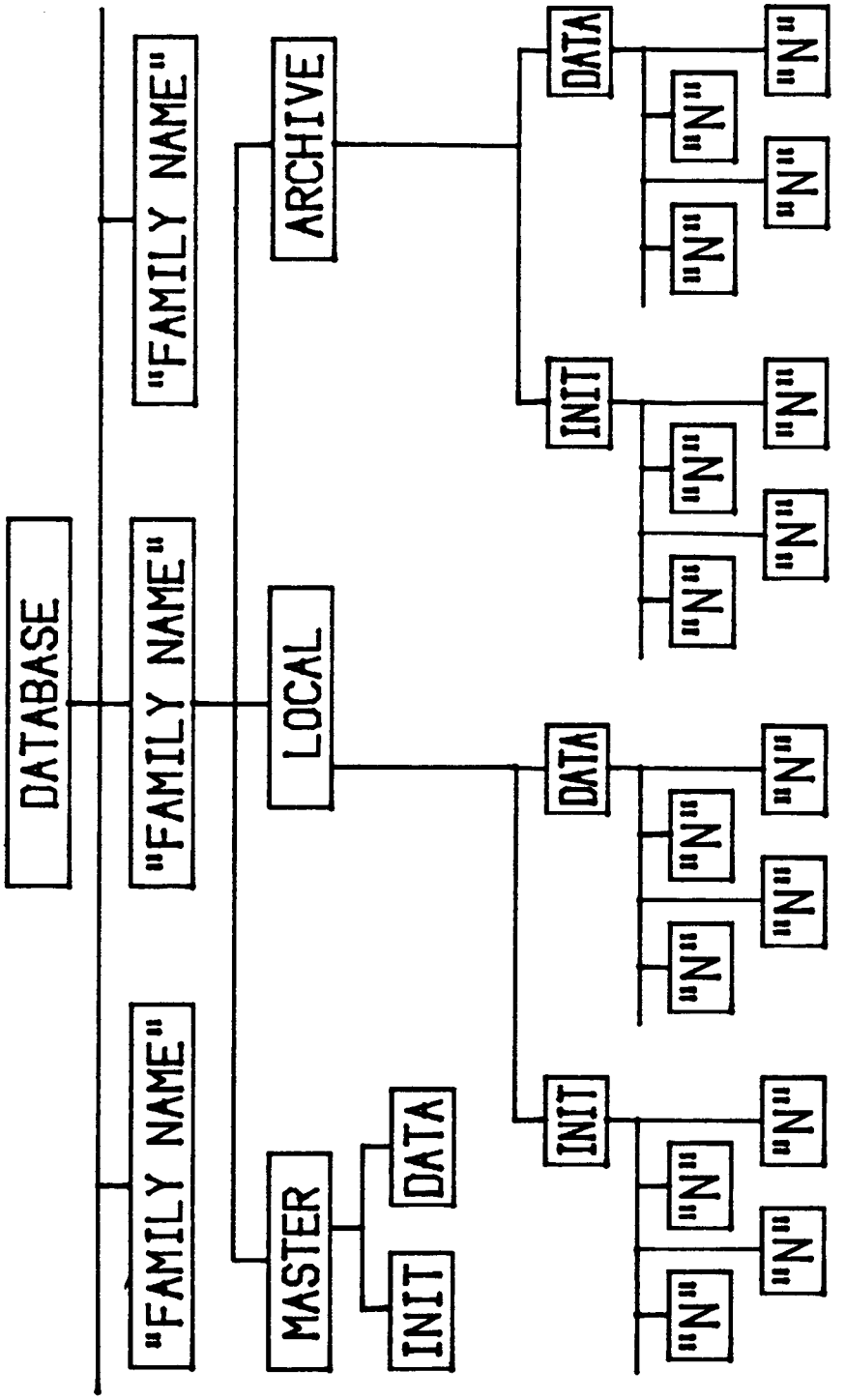
Disc Library/File Structure for the RECORDKEEP II Software

TITLE

RECORDKEEP II

ABSTRACT NO:

51/07-6108/1



Disc Library/File Structure for the Databases used
by RECORDKEEP II

"N" are names of databases created by the user.

"FAMILY NAME" are database Family Names assigned by the user when he created a new MASTER database.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

REFERENCES

This software was derived from a program documented in section 16, pages 4-212 to 4-219 of the PLOT50 Business Planning and Analysis Manual. Substantial changes have been made to store databases on a 4907. The instructions in the above manual are no longer applicable.

The sorting method used in command was taken from the following source:

Knuth, Sorting & Searching - The Art of Computer Programming. Addison - Wesley Publishing Company, 1975 pp114-123.

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

APPENDIX A

RECORDKEEP II is a general purpose program that can be applied to many applications, but in order to fit a particular application well it may need more specific tasks to be added. Because of the manner in which RECORDKEEP II is designed, it is easy to add more tasks and there are few practical limits to the number that can be added. This Appendix will describe the procedure for adding new tasks and the programming conventions that must be followed.

RECORDKEEP II is composed of parts, a main program and many program segments called overlays. Overlaying is a method where many program segments share one area in memory. This is possible because the main program controls which segment is in memory, depending on the specific task it must perform The secret to understanding how the program can select specific overlays is to understand some common properties all overlays have. Overlays begin in the same manner. The line numbers begin at 4000 and the first line is always a REMARK statement. This statement is used as a target line when the overlays are appended into memory. The range of line numbers are also limited. They must begin at 4000 and never exceed 8999. This insures that the overlays fit within the main program. The other properties overlays have in common are their usage of variables. Variables fall into two groups, local variables and global variables. Global variables are used to communicate between the overlays and main program, while the local variables are reserved for the exclusive use of the overlay in memory. The file names for all overlays are similar too. Overlay file names are found in the same libraries (@DBMPROG/RECORDKEEP/) and begin with the letters

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

"OV". Individual overlays are differentiated by a numeric suffix on the end of the file name and the main program appends in any given overlay by concantinating the appropriate suffix to the character string "@DBMPROG/RECORDKEEP/OV" The overlay stored on disc under this file identifier is then appended into memory. Suffixes may also end in a letter (e.g. OV3A, OV7A), but these overlays are called (meaning selected to be appended into memory next) from other overlays.

The user drives RECORDKEEP II from a Menu by entering in a Menu number that corresponds to a particular task. Once the user selects a task the program must append into memory the overlay that contains the task. This linkage between a menu selection, and the right overlay is done with a "table look-up". The global variable A is a table of overlay suffixes. When the user selects a menu number the program "looks-up" the suffix stored in the element of A equal to the menu number. If the user selected menu item 3 then the overlay that contains that task has a suffix equal to A(3). The main program then appends in the appropriate overlay and transfers control to the overlay. The overlay performs the task the user selected and transfers control back to the main program by jumping to line 9000. Once the overlay relinquishes control back to the main program all disc files should be closed.

There are few practical limits to how many additional tasks can be added. The two constraints are available disc space to hold programs and overlays and the amount of available memory in the 405X running the program. Available disc space is

JG/3A/11

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

ample for a great number of additional overlays but the 405X memory constraints can pose a problem at times. RECORDKEEP II is designed to run on a machine with a minimum 16K of memory. For people with larger memory in their 405X each existing overlay in RECORDKEEP II can be enlarged with new tasks as long as the overlays don't violate the rules discussed above and they fit in memory. For tasks that cannot fit into memory the task can be split into many overlays, with the first overlay calling each subsequent overlay. An example of this are overlays OV3 and OV3A. Notice that overlays that are called by other overlays end in a letter. This helps to clarify which overlays are related.

When adding a new task the first decision is whether to include the task in an existing overlay or to write a new overlay. This Appendix will cover writing a new overlay, modifying a existing overlay is a blend of normal program editing and techniques used to write a new overlay. The best way to learn how to write a compatible overlay is to examine the existing overlays in RECORDKEEP II. Overlays can be written in a modular manner because certain utility subroutines needed to open files and to verify user inputs can be borrowed from other overlays and used as building blocks for the new overlay. These utilities are covered in detail in Appendix B. To begin, consider a hypothetical task to be added to RECORDKEEP II. This task will calculate total cost of units in an inventory by reading the number of units on hand and the unit price from the MASTER database, multiply together these two numbers and store the result (total cost) in an existing numeric field in the MASTER database. It is assumed that a RECORDKEEP II MASTER database exists, the number of units begins in column 12 and is 5 characters long,

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

unit price begins in column 23 and is 6 characters long and the total cost field begins in column 60 and is 10 characters long. This defines the task, and it can be implemented in 3 steps. The first step is to write the overlay, which must conform to the programming conventions mentioned in this document. The second step is to modify the main progam and overlay OVO. These modifications will allow the new overlay to be linked to an entry in the menu of the main program, giving the user control over when it will be executed. The last step requires changes to the "AUTO LOAD" Program. These changes will allow the backup copy of the overlay that is saved on tape to be automatically copied to disc from the menu of the "AUTO LOAD" Program. As changes are made to each program be sure to save the program on the program tape. The order that the various programs reside on tape is very important so great care should be taken to preserve this order when saving programs. Once all three steps have been performed the new version of software can be updated on a program disc via menu selection 3 of the "AUTO LOAD" program. It would also be wise to write to protect the program tape, since it is the software backup in emergencies.

JG/3A/13

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

WRITING THE OVERLAY

All overlays begin at line number 4000 and never exceed line number 8999. Line 4000 is always a remark statement, because this line will be the target line for later appends. Look at the source listing for OV5, it will be a good model. OV5 begins with the remark statement on Line 4000 and the next few lines document what overlay it is. Following these remark statements are a series of DELETE statements in lines 4060 through 4100. These should be included in all overlays except overlays called by other overlays because they set the local variables to a known state. Variables not included in these DELETE statements are considered Global variables which pass information between overlays. Global variables should be changed with great care since they may affect other overlays. Local variables do not affect other overlays, but the utility subroutines described in Appendix B use local variables so plan ahead to avoid any local variable conflicts with these utilities. OV5 actually contains three tasks, and the next six lines direct the program flow to the appropriate subroutine, depending on the menu number stored in A3 that the user selected. Below is an alternative way that saves 5 lines of code.

```
GOSUB  1 + (A3=8) + (A3=9)*2  OF 4180, 4310, 4490
```

In OV5 each task is treated as a subroutine and when it returns from a task subroutine, line 4170 returns control to the main

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

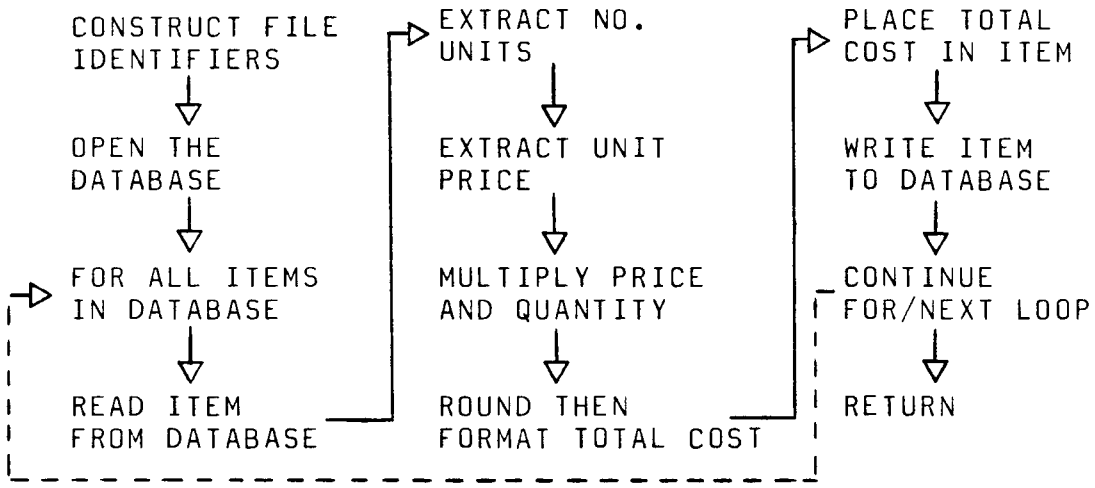
program by jumping to line 9000. If only one task is in the new overlay then the subroutine selection can be left out. When transfering control back to the main program it is assumed that all files have been closed. This can be handled by the task subroutine or done just before control is transfered. By following the suggestions above the first part of the example overlay would look like the following.

```
4000 REM          TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------
4030 REM
4040 REM                  OVERLAY            OV8
4050 REM
4060 DELETE   H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE   I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE   J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE   K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE  I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$
4110 GOSUB   4150
4120 CLOSE
4140 GO TO   9000
4150 REM          TASK TO COMPUTE UNIT TYPE COST
4160  "
      "
      "
      "
```

In the example above the task was converted to a subroutine. This is not the most efficient way to write the overlay, but it

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

has the advantages of being more modular. This will make it easier to add tasks in the future. To this point the new overlay is very general and would apply to any task. The task specific portion of the overlay will be written next. The flow chart below outlines what has to be done:



The following description explains the new overlay flow charted above. The actual overlay can be found on the following pages.

The file identifiers can be constructed using a utility sub routine described in Appendix (B). This utility receives the database name in L$ and returns the file identifiers for the INIT file in E$ and the file identifier for the DATA file in F$. When L$ contains a null string then the file identifier for the MASTER database are returned. E$ and F$ are then used by another utility subroutine to open up the databases. Once the databases are open the program proceeds to read in an item, perform the necessary computations, write the item back with total cost inserted. When all items are processed the files are closed and control is passed back

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4000 REM          TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                        OVERLAY     OV8
4050 REM
4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$
4110 GOSUB 4150
4120 REM       CLOSE ALL FILES AND RETURN TO MAIN PROGRAM
4130 CLOSE
4140 GO TO 9000
4150 REM       TASK TO COMPUTE TOTAL COST
4160 L$=""
4170 REM       CONSTRUCT FILE IDENTIFIERS
4180 GOSUB 4830
4190 REM       OPEN DATABASES
4200 GOSUB 4650
4210 DIM U$(T4)
4220 REM       FOR ALL ITEMS IN DATABASE
4230 FOR I=1 TO T2
4240 REM       READ ITEM FROM DATABASE
4250 READ #2,I:U$
4260 REM       EXTRACT NUMBER OF UNITS
4270 R$=SEG(U$,12,5)
4280 H1=VAL(R$)
4290 REM       EXTRACT UNIT PRICE
4300 R$=SEG(U$,23,6)
4310 H2=VAL(R$)
4320 REM       MULTIPLY PRICE AND QUANITY
4330 H3=H1*H2
4340 REM       ROUND H3 TO HUNDREDTHS THEN ADD .001
4350 REM       FOR FORMATTING PURPOSES
4360 GOSUB 4610
4370 REM       CONVERT H3 TO A STRING AND RIGHT JUSTIFY
4380 REM       IF H3 >= TO 10^7 THEN IT WONT FIT,GIVE ERROR
4390 IF H3<10^7 THEN 4420
4400 PRINT "CALCULATED COST EXCEEDED ITEM FIELD WIDTH"
4410 STOP
4420 DELETE R$
4430 DIM R$(12),Q$(10)
4440 IF H3<>1.0E-3 THEN 4470
4450 Q$="      0.00"
4460 GO TO 4530
4470 R$=STR(H3)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4480 R$=REP("",1,1)
4490 Q$=SEG(R$,1,LEN(R$)-1)
4500 IF LEN(Q$)=10 THEN 4530
4510 R$=SEG(X$,1,10-LEN(Q$))
4520 Q$=R$&Q$
4530 REM       PLACE TOTAL IN ITEM
4540 U$=REP(Q$,60,10)
4550 REM       WRITE ITEM TO DATABASE
4560 WRITE #2,I:U$
4570 REM       CONTINUE FOR/NEXT LOOP
4580 NEXT I
4590 REM       RETURN
4600 RETURN
4610 REM       ROUND H3 TO CENTS THEN ADD .001
4620 H4=ABS(H3)+1.0E-12
4630 H3=(INT(H4*100+0.5)/100+1.0E-3)*((H3<0)*-1+(H3=>0)*1)
4640 RETURN
4650 REM   ***   OPEN DATABASE FILES
4660 CLOSE
4670 E0=0
4680 GOSUB 4830
4690 CALL "FILE",A9,E$,K$
4700 IF LEN(K$)>0 THEN 4750
4710 E0=1
4720 PRINT "↳*↳*** File was not found on disc.↳*"
4730 PRINT USING "4(110(""↳"")),""↳↳↳""":
4740 GO TO 4820
4750 OPEN E$;1,"F",K$
4760 OPEN F$;2,"F",K$
4770 READ #1:T1,T2,T9,T8,T4,S1
4780 DELETE C,T$
4790 DIM C(T1,2),T$(T4)
4800 READ #1:C,T$
4810 S$=SEG(X$,1,T8)
4820 RETURN
4830 REM    SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
4840 REM   L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.
4850 K$="/LOCAL/INIT/"
4860 K$=REP(L$,LEN(K$)+1,0)
4870 IF L$<>"" THEN 4900
4880 K$=REP("MASTER",2,5)
4890 K$=SEG(K$,1,LEN(K$)-1)
4900 E$=G$&K$
4910 K$="/LOCAL/DATA/"
4920 K$=REP(L$,LEN(K$)+1,0)
4930 IF L$<>"" THEN 4960
4940 K$=REP("MASTER",2,5)
4950 K$=SEG(K$,1,LEN(K$)-1)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4960 F$=G$&K$
4970 RETURN
```

NOTE: The following characters appear differently on a 4050 Series Graphic System.

"␈" is a CONTROL H
"␋" is a CONTROL K
"␡" is a CONTROL-SHIFT RUBOUT
"␊" is a Line Feed
"^" is an up arrow (exponentiation)

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

to the main program. Processing is begun by reading each item into a character string. The composition of the item dictates where the number of units, unit price and total field are found. For the example some arbitrary beginning positions and field lengths were assumed and they are summarized in the table below:

	Beginning Position	Field Length
Number of units:	12	5
Unit price	23	6
Total cost	60	10

The number of units are extracted from U$ and then converted to a number held in H1. In a similar fashion the unit price is stored in H2. Multiplying H1 times H2 to yield the total cost (H3) is not difficult, but before H3 is placed back into U$ several formating and error checks are performed. H3 is calculated and rounded to the nearest cent, and a small constant was added to make sure that at least two decimal places will always be present. Next H3 is checked to be sure that the number can fit in the space alloted for the total cost field. H3 is converted to a string (R$) and then the first and last characters are removed. The first character is always a blank, due to the STR function. The last character is not wanted because it results from adding .001 to H3. The resulting string stored in Q$. The number Q$ contains will have two

JG/3A/17

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

places to the right of the decimal place. All that is left to be done is to be sure the length of Q$ is equal to 10, if not, Q$ is padded with blanks (right justified) so it is. Q$ can now be placed in the total field of the item and the item written back into the database. This covers the basics of how to write an overlay and an example of how one would look. The majority of the work has been done, but changes have to be made to the main program and OVO to link the new overlay to the program as a whole. RECORDKEEP II can automatically copy to disc the backup copies of the software. To include the new overlays in this automatic copy feature requires changes to be made to the "AUTO LOAD" program.

LINKING NEW OVERLAYS

The menu in the main program must be expanded to include the new task, and the error trap that filters incorrect menu selections must reflect the new menu number. The main program must never be renumbered. Line numbers 300 through 430 are reserved for the print statements that produce the menu. It is permissable to change the increment between line numbers, this would permit a maximum of 149 tasks in the menu, and still have the menu between lines 290 and 430. Overlay OVO also requires a change. OVO is where the global variable A is initialized. Since A is a table of suffixes it must be updated to reflect the suffix of the overlay that contains the new task. For the example cited above the required changes are outlined below. The first change is to include a description of the new task in the main menu. Since line 420 contains the last menu item,

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

line 421 is where the new print statement is added. The last menu number is 12 so the new task will have a menu number of 13.

```
421 PRINT "JI13.  CALCULATE UNIT COST * # UNITS"
```

Line 450 is the error trap. Valid menu numbers are now 1 through 13 so it must be changed to allow the new menu number to pass.

```
450   IF   A3<=0   OR   A3>13   THEN   290
```

These are the only changes that need to be made to the main program, but one change remains to be made to overlay OVO. To complete the linkage global variable A must be updated to contain the appropriate overlay suffix. To generalize the discussion, call the menu number of the new task "I". Line 4690 in overlay OVO is a data statement containing the suffixes to be stored in "A". Edit this data statement so that for each new task added to the menu, the Ith element of A [A(I)] will be equal to the overlay suffix that contains the task. If the task was included in an existing overlay then A(I) would equal the suffix of this overlay, but in the example cited above a new overlay was written and its suffix was 8. Therefore, the 13th number in the data statement should be set to 8 as shown below,

```
4690   DATA   1,1,2,6,3,4,4,5,5,7,6,5,8,0,0,0,0,0,0,0
```

JG/3A/19

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

Now all the necessary changes have been made so that when the user selects menu number 13, overlay OV8 will be appended into memory. The remaining step is to change the "AUTO LOAD" program to load any new overlays onto disc with the other system programs.

LOADING NEW OVERLAYS

The "AUTO LOAD" program has a menu of three items. The third item will transfer all the programs, this would include overlays, from the program tape to the disc selected to hold the programs. Any new overlays will not be included in this transfer unless the program is changed to include them. There are three variables that control the transfer. A$ contains all the program names. Any new overlay must have its name added to A$. Variable A holds the beginning position for the file name, in the order they appear on the tape, and starting with program MAIN (file 3). Variable B holds the length of the file names and are in the same order as the values in A. The number of files transfered to the disc are controlled in a FOR/NEXT loop starting at line 30360. If the new tasks were added to existing overlays then the "AUTO LOAD" program does not need to be changed and this step can be ignored. When a new task requires a new overlay then the following changes must be made to the "AUTO LOAD" program. Line 30220 is where A$ is defined. Add the new overlay file name (e.g. OV8) to A$. Next change the data statements in lines 30240 and 30250. If the overlay was saved on file 14 on the program tape then it would be the 12th file after program FILEINIT. The 12th number in the data statement on line 30250 should be changed to the character

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

length of the file name of the overlay in file 14. The interaction between the contents of A$, A, B and the order that the programs occur on the tape can be confusing. If necessary trace the program flow to verify how they work together.

Below is what A$, A and B should be changed to if a new overlay called OV8 is to added to the loading sequence.

```
30220 A$="MAIN OV0  OV1  OV2  OV3  OV4  OV5  OV6  OV7  OV7A OV8"
30230 RESTORE  30240
30240  DATA  1,10,14,18,22,27,31,35,39,43,6,48,0,0,0,0,0,0,0,0
30250 DATA 4,3,3,3,4,3,3,3,3,4,3,3,0,0,0,0,0,0,0,0
```

The remaining change is to alter the range of tape files transfered. This is done by changing the FOR/NEXT statement in line 30360. The number following the "TO" in this statement should be the file number of the last tape file that contains a program. In the context of the example, overlay OV8 would be saved in file 14 and, therefore, line 30360 would be changed to:

```
30360  FOR  I=3  TO  14
```

Be careful not to renumber the program when the changes are being made. The "AUTO LOAD" program should never be renumbered. With that one precaution understood the edited "AUTO LOAD" program can be saved on file 1 of the program tape.

JG/3A/21

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

All three steps have been described and with this documentation adding tasks to RECORDKEEPII should not be too difficult. Some of the basic building blocks can be found in the Appendix B, which can perform some of the common utility functions in new overlays. Once the overlays are written, only a few changes in other programs incorporate the new overlay into the collection of interacting programs called RECORDKEEP II.

JG/3A/10-22

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

APPENDIX B

Writing a new overlay is not difficult, but there is no point in investing time to solve problems, which already have a solution. The following utility subroutines will help reduce the time it takes to write an overlay by offering solutions to some of the more common task overlays must do. Task such as opening data bases, verifying and formating keyboard entries, and selecting 4054 character sizes. Several of these tasks require more than one utility subroutine, but they work together, each depending on the other to do what it can't. Subroutines 1 through 3 are used to open data bases. Subroutine 1 verifies that the requested data base name is a valid name, while subroutine 2 takes the name and constructs file identifiers (FI) for the pair of data bases. The databases are actually opened by subroutine 3. Keyboard entries that enter the database are verified and padded with subroutines 4 and 5. Alphanumeric inputs are handled by subroutine 4 and numeric entries use subroutine 5. Subroutine 6 is actually two subroutines, one that selects the appropriate character size on a 4054, and another which resets the character size.

Using these subroutines will hopefully reduce the time it takes to produce a working overlay.

JG/A/27

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SUBROUTINE 1

Any time an operator keys in a database name it is best to verify that the name has the proper syntax. This will avoid unrecoverable errors later on. For the sake of simplicity RECORDKEEP II does not allow file extensions, but other than that exception this subroutine will check a file or library name in L$ for the proper syntax. The name must be 10 characters or less, start with a letter, and contain only letters and numbers, imbedded blanks are not allowed. If the name passes the test variable E0 will be equal to zero when control is returned to the calling program. When the syntax is incorrect E0 is set to 1 and a diagnostic error message is printed, before control is returned.

VARIABLE	TYPE	DESCRIPTION
E0	local	Error flag E0=0 then no error. E0=1 then the name is illegal.
I0	local	FOR/NEXT index
K6	local	Scratch variable
L$	local	Contains the name to be tested
P$	local	Holds one character of L$ to be examined

JG/A/28

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
6330 REM        CHECK THE PASSWORD OR FILE NAME IN L$
6340 E0=0
6350 IF LEN(L$)<11 THEN 6380
6360 PRINT "*↯*** The name is longer than ten characters. Try again.↯*"
6370 GO TO 6520
6380 P$=SEG(L$,1,1)
6390 K6=ASC(P$)
6400 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 6430
6410 PRINT "*↯*** The name does not start with a letter. Try again.↯*"
6420 GO TO 6520
6430 FOR I0=1 TO LEN(L$)
6440 P$=SEG(L$,I0,1)
6450 K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
6460 K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
6470 IF K6<=0 THEN 6500
6480 NEXT I0
6490 GO TO 6530
6500 PRINT "*↯*** The name contains blanks or non alphanumeric ";
6510 PRINT "characters.↯*** Try again.↯*"
6520 E0=1
6530 RETURN
```

NOTE: The following characters appear differently on a 4050 Series Graphic System.

"↯" is a CONTROL-SHIFT RUBOUT

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SUBROUTINE 2

Databases in RECORDKEEP II fall into two classes. The INIT class contains the quantitatitive description of the second class of database. The second class, the DATA class, contains all the items in the database, this one is most visable to the user. The purpose of this subroutine is to take a name stored in L$ and construct the appropriate file identifiers (FI) for each class of database. If L$ is a null string, the file identifiers for the "MASTER" database are returned. When contol is passed back to the calling program E$ will contain the F.I. for the INIT class database and F$ will have the F.I. for the DATA class database. No error traps are used because the syntax of the name in L$ is assumed to be correct.

VARIABLE	TYPE	DESCRIPTION
E$	Global	F.I. for the INIT class database
F$	Global	F.I. for the DATA class database
G$	Global	The library structure containing the database family name. This is used as a prefix when constructing the F.I.'s
K$	Local	Scratch variable
L$	Local	The database name to be incorporated into the F.I.'s

JG/A/29

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4830 REM    SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
4840 REM   L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.
4850 K$="/LOCAL/INIT/"
4860 K$=REP(L$,LEN(K$)+1,0)
4870 IF L$<>"" THEN 4900
4880 K$=REP("MASTER",2,5)
4890 K$=SEG(K$,1,LEN(K$)-1)
4900 E$=G$&K$
4910 K$="/LOCAL/DATA/"
4920 K$=REP(L$,LEN(K$)+1,0)
4930 IF L$<>"" THEN 4960
4940 K$=REP("MASTER",2,5)
4950 K$=SEG(K$,1,LEN(K$)-1)
4960 F$=G$&K$
4970 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SUBROUTINE 3

This subroutine will open the files with F.I. E$ and F$. If these files can not be found on the database disc an error message is logged and the subroutine is exited with K0 equal to 1, an error flag. Normally the files do exist, and the INIT, and DATA class databases are opened. The subroutine then reads all the quantitative information off the INIT class database and exits with K0 equal to zero, indicating no errors.

VARIABLE	TYPE	DESCRIPTION
A9	Global	Unit number of the drive containing the database disc
C	Global	See page 32
K0	Local	Error flag. K0=0 then no error. K0=1 the files not opened
S1	Global	See page 32
T1	Global	See page 32
T2	Global	See page 32
T4	Global	See page 32
T8	Global	See page 32
T9	Global	See page 32
E$	Global	F.I. for the INIT class database
F$	Global	F.I. for the DATA class database

JG/3A/30

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

VARIABLE	TYPE	DESCRIPTION
K$	Local	Scratch variable
S$	Global	String of blanks T8 characters long
T$	Global	See page 32
X$	Global	String of blanks 135 characters long

```
4650 REM   ***   OPEN DATABASE FILES
4660 CLOSE
4670 E0=0
4680 GOSUB 4830
4690 CALL "FILE",A9,E$,K$
4700 IF LEN(K$)>0 THEN 4750
4710 E0=1
4720 PRINT "↓*↓*** File was not found on disc.↓*"
4730 PRINT USING "4(110(""↓""")),""↓↓↓""":
4740 GO TO 4820
4750 OPEN E$;1,"F",K$
4760 OPEN F$;2,"F",K$
4770 READ #1:T1,T2,T9,T8,T4,S1
4780 DELETE C,T$
4790 DIM C(T1,2),T$(T4)
4800 READ #1:C,T$
4810 S$=SEG(X$,1,T8)
4820 RETURN
```

NOTE: The following characters appear differently on a 4050 Series Graphic System.

"↓" is a CONTROL H
"↓" is a CONTROL K
"↓" is a CONTROL-SHIFT RUBOUT
"↓" is a Line Feed

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SUBROUTINE 4

Before a string, destined for a alphanumeric field, can be placed in an item it must be left justified and padded with blanks so that the resulting string is as long as the field is wide. This subroutine does just that. O$ enters the routine an arbitrary length, but less or equal to the field width I3. When control is returned to the calling program O$ has a length equal to I3.

VARIABLE	TYPE	DESCRIPTION
I3	Local	The width of field
O$	Local	The original string, which is padded to be I3 characters long, and left justified
W$	Global	Scratch variable
X$	Global	String of blanks 135 characters long

```
5310 REM PACK W/SPACES-ALPHANUMERIC
5320 W$=SEG(X$,1,I3-LEN(O$))
5330 O$=O$&W$
5340 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SUBROUTINE 5

RECORDKEEP II stores numbers in numeric fields as strings, and it assumes that the string contains a valid numeric constant. This subroutine examines a string 0$ and determines if it contains a valid numeric constant. If it does 0$ is returned as a string 13 characters long, with the constant right justified in the string. If not it prints a dianostic error message and sets K0 equal to 1. This subroutine is more forgiving than the BASIC interpreter. Any blanks or commas in 0$ are deleted before 0$ is returned. This subroutine will accept constants in an integer, real or scientific notation format.

VARIABLES	TYPE	DESCRIPTION
H1	Local	Length of 0$
H2	Local	FOR/NEXT index. Identifies character in 0$ to be processed
H4	Local	Length of J$
K0	Local	Error flag. K0=0 then no errors. K0=1 then 0$ contains an illegal numeric constant
K1	Local	A count of the number of "E's" in 0$
K2	Local	A count of the number of "."'s in 0$

JG/3B/3

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

VARIABLES	TYPE	DESCRIPTION
K3	Local	A flag that is set (K3=1) when the first numeric character is encountered in the mantissa or exponent of the constant
K4	Local	The class of the character being processed (Y$)
J$	Local	The reconstructed string containing the numeric constant, built as each character in O$ passes inspection
O$	Local	The string that contains the constant to be tested
V$	Local	The last valid character previous to the character being processed (Y$)
W$	Global	Scratch variable
X$	Global	String of blanks 135 characters long
Y$	Global	Scratch variable. Contains the current character being processed

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4800 REM    VERIFY THE STRING IS NUMERIC THEN RIGHT JUSTIFY
4810 RESTORE 4820
4820 DATA 0,0,0,0,0,0
4830 READ K0,H4,K1,K2,K3,K4
4840 DELETE J$,V$
4850 H1=LEN(O$)
4860 DIM J$(H1),V$(1)
4870 J$=""
4880 V$=""
4890 FOR H2=1 TO H1
4900 Y$=SEG(O$,H2,1)
4910 K4=POS("0123456789.+-E, ",Y$,1)
4920 IF K4<=0 THEN 5280
4930 REM   CASE 1 -  Y$= A COMMA OR BLANK   "IGNORE IT"
4940 IF K4=15 OR K4=16 THEN 5040
4950 IF K4=14 THEN 5120
4960 IF K4=12 OR K4=13 THEN 5190
4970 IF K4=11 THEN 5230
4980 REM   Y$ IS A NUMERIC CHARACTER
4990 K3=1
5000 REM   ADD CHARACTER TO OUTPUT STRING (J$) AND INC. COUNTER (H4)
5010 H4=H4+1
5020 J$=REP(Y$,LEN(J$)+1,0)
5030 V$=Y$
5040 NEXT H2
5050 REM   CHECK J$ FOR ZERO LENGHT OR NO DIGITS
5060 IF K3=0 OR LEN(J$)<=0 THEN 5280
5070 REM   PACK NUMERIC STRING RIGHT JUSTIFIED
5080 O$=J$
5090 W$=SEG(X$,1,I3-LEN(O$))
5100 O$=REP(W$,1,0)
5110 RETURN
5120 REM   CASE 2 -  Y$= "E"
5130 IF LEN(V$)=0 THEN 5280
5140 IF ASC(V$)=43 OR ASC(V$)=45 THEN 5280
5150 K1=1+K1
5160 IF K1>1 THEN 5280
5170 K3=0
5180 GO TO 5000
5190 REM   CASE 3 -  Y$= A PLUS OR MINUS SIGN
5200 IF V$="" THEN 5000
5210 IF ASC(V$)<>69 THEN 5280
5220 GO TO 5000
5230 REM   CASE 4 -  Y$= A DECIMAL POINT
5240 K2=K2+1
5250 IF K2>1 THEN 5280
5260 IF K1=>1 THEN 5280
5270 GO TO 5000
5280 PRINT "↳**↳***  Illegal numeric entry. Try again.↳*"
5290 K0=1
5300 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

SUBROUTINE 6

When a 4054 is used to display a database with a width greater than 72, it can compensate by reducing its character size. Any database with a width (T4) less than 133 characters can be viewed on the screen of the 4054 without "wraparound". Each subroutine determines if a 4054 is being used by the value of RND(0). If a 4051 or 4052 is being used or output is directed to a printer control is quickly passed back to the calling program, if not the appropriate character size is choosen, based on T4, and the variable A3 is changed to the number of lines that can fit on the screen. The second subroutine returns the 4054 back to character size 4 and 30 lines per page. Some overlays need to view only one item from the database at a time. In these cases the subroutines have been modified by removing the check for printer output, deleting the lines referencing variable A2, and explicitly using primary adress 32 in all print statements.

VARIABLE	TYPE	DESCRIPTION
A1	Global	Primary address where reports generated in overlay OV6 are printed
A2	Global	Number of lines per page that fit onto one page of the device at primary address A1
K9	Local	The 4054 character size that will prevent "wraparound" on the screen
T4	Global	See page 32

JG/B/5

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

This set of subroutines are used when
printing reports in OV6

```
6540 REM    SELECT CHARACTER SIZE
6550 IF RND(0)<0.8 THEN 6600
6560 IF A1<>32 THEN 6600
6570 K9=1+(T4-73<0)+(T4-80<0)+(T4-120<0)
6580 A2=(K9=1)*56+(K9=2)*54+(K9=3)*34+(K9=4)*30
6590 PRINT @A1,17:K9
6600 RETURN
6610 REM    RESET CHARACTER SIZE
6620 IF RND(0)<0.8 THEN 6660
6630 IF A1<>32 THEN 6660
6640 PRINT @A1,17:4
6650 A2=30
6660 RETURN
```

This set of subroutines are used when
printing only one item to the screen

```
6420 REM    SUBROUTINE TO SELECT CHARACTER SIZE ON 4054
6430 IF RND(0)<0.8 THEN 6460
6440 K9=1+(T4-73<0)+(T4-80<0)+(T4-120<0)
6450 PRINT @32,17:K9
6460 RETURN
6470 REM    SUBROUTINE TO RESET CHARACTER SIZE ON 4054
6480 IF RND(0)<0.8 THEN 6500
6490 PRINT @32,17:4
6500 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 1

```
1 REM ***  WARNING:  This program is sensitive to being renumbered.
2 REM ***            Any changes in line numbers of this program
3 REM ***            may require changes to be made to other programs.
4 REM
5 REM       SET CHARACTER SIZE ON 4054

6 IF RND(0)<0.8 THEN 100
7 PRINT @32,17:4

8 REM
93 REM *************************
94 REM ***  51/07-6108/1    ***
95 REM ***  RECORDKEEP II   ***
96 REM ***                  ***
97 REM ***                  ***
98 REM *************************

100 INIT
110 PRINT USING 120:
120 IMAGE "LJJJ",16X,"Data Base Management and Report GenerationJ"
130 CALL "TIME",A$
140 IF LEN(A$)>0 THEN 190
150 PRINT "JPlease enter the current date and time as"
160 PRINT "(DD-MMM-YY HH:MM:SS): ";
170 INPUT A$
180 CALL "SETTIM",A$
190 PRINT "JThe disc with the programs will be placed in which";
200 PRINT " device? ";
210 INPUT A8
220 IF A8=>0 AND A8<3 THEN 280
230 PRINT USING 240:
240 IMAGE "J* ***  The device address you entered above does not",S
250 PRINT USING 260:
260 IMAGE " equal 0, 1, or 2._*"
270 GO TO 190
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
280 PRINT "JThe disc with the database will be placed in which";
290 PRINT " device? ";
300 INPUT A9
310 IF A9=>0 AND A9<3 THEN 350
320 PRINT USING 240:
330 PRINT USING 260:
340 GO TO 280
350 IF A8=A9 THEN 450
360 PRINT "JPlease insert the program disc in device ";A8;"."
370 PRINT "Then press Return."
380 INPUT A$
390 CALL "MOUNT",A8,A$
400 PRINT "JPlease insert the database disc in device ";A9;"."
410 PRINT "Then press Return."
420 INPUT A$
430 CALL "MOUNT",A9,A$
440 GO TO 490
450 PRINT "JPlease insert the program/database disc in device ";A9;"."
460 PRINT "Then press Return."
470 INPUT A$
480 CALL "MOUNT",A9,A$

490 REM          BEGIN OF MENU

500 PRINT USING 120:
510 PRINT USING """JI"",18X,""MENU""":
520 PRINT "I1)  Create a new master database?"
530 PRINT "I2)  Manipulate an existing master database?"
540 PRINT "I3)  Load system programs onto disc?"
550 PRINT "JPlease enter your selection: ";
560 INPUT A
570 IF A>0 AND A<4 THEN 600
580 PRINT "J*  ***  Your selection is ot in the Menu. Try again.  *"
590 GO TO 550
600 GOSUB A OF 30500,30530,30000
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
30000 REM
30010 REM
30020 REM
30030 REM
30040 REM
30050 REM
30060 REM
30070 REM
30080 REM
30090 REM
30100 REM
30110 REM
30120 REM
30130 REM
30140 REM
30150 REM
30160 REM
30170 REM       OVERLAY MAIN PROGRAM TO TRANSFER SYSTEM PROGRAMS TO DISC
30180 REM

30190 DELETE A,B,A$,B$,C$,D$
30200 DIM A(20),B(20)
30210 CALL "UNIT",A8
30220 A$="MAIN OV0 OV1 OV2 OV3 OV3A OV4 OV5 OV6 OV7 OV7A"
30230 RESTORE 30240
30240 DATA 1,10,14,18,22,27,31,35,39,43,6,0,0,0,0,0,0,0,0,0
30250 DATA 4,3,3,3,4,3,3,3,3,4,3,0,0,0,0,0,0,0,0,0
30260 READ A,B
30270 DELETE 1,29999
30280 FIND 2
30290 D$="FILEINIT"
30300 B$="@DBMPROG/FILEINIT"
30310 H9=MEMORY
30320 APPEND 30000,0
30330 SAVE B$;1,29999
30340 PRINT "JJ*** ";D$;" TRANSFERED ***"
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
30350 B$="@DBMPROG/RECORDKEEP/"
30360 FOR I=3 TO 13
30370     I1=I-2
30380     DELETE 1,29999
30390     RENUMBER 30000
30400     FIND I
30410     H9=MEMORY
30420     APPEND 30000,0
30430     C$=SEG(A$,A(I1),B(I1))
30440     D$=B$&C$
30450     SAVE D$;1,29999
30460     PRINT "*** ";C$;" TRANSFERED ***"
30470 NEXT I
30480 PRINT "J***  Program transfer complete.  ***"
30481 PRINT "JJJPress AUTO LOAD key to select other programs."
30490 END

30500 REM      CALL IN FILEINIT AS AN OVERLAY

30510 B$="@DBMPROG/FILEINIT"
30520 GO TO 30550

30530 REM      CALL IN MAIN AS AN OVERLAY

30540 B$="@DBMPROG/RECORDKEEP/MAIN"
30550 CALL "FILE",A8,B$,A$
30560 CALL "UNIT",A8
30570 IF LEN(A$)>0 THEN 30620
30580 PRINT "J* ***  Program was not found on the disc. Have you loaded"
30590 PRINT "***  the system programs onto the disc?_*"
30600 PRINT USING "5(110(2("" H"""))),S":
30610 GO TO 490
30620 DELETE 1,29999
30630 H9=MEMORY
30640 APPEND B$;30000,0
30650 GO TO 1
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
30660 END
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 2

```
1 REM
100 REM      ***   FILEINIT
110 REM      ***   THIS PROGRAM INITIALIZES THE DATABASES USED BY
120 REM      ***   RECORDKEEP. THE PROGRAM WAS DERIVED FROM A PROGRAM
130 REM      ***   DOCUMENTED IN SECTION 16 PP 4-212 TO 4-219 OF THE
140 REM      ***   PLOT50 BUSINESS PLANNING AND ANALYSIS MANUAL.
150 REM      ***   SUBSTANTIAL CHANGES HAVE BEEN MADE TO STORE THE
160 REM      ***   DATABASES ON A 4907 FLEXIBLE DISC SO THESE INSTRUC-
170 REM      ***   TIONS ARE NO LONGER APPLICABLE.
180 REM      ***

190 DELETE 30000,40000
200 PRINT "LIFILE INITIALIZATION ROUTINE"
210 PRINT "JThis routine will ask for information regarding titles, ";
220 PRINT "size and the number of fields required for each item. ";
230 PRINT "The  first field for"
240 PRI "each item will be a 5-character record number in the file and
250 PRINT "labeled as 'ITEM#'."
260 PRINT "JHow many additional fields of information will be needed"
270 PRINT "for each item:  ";
280 INPUT T1
290 T1=T1+1
300 PRINT "JEnter the width (in characters) of the media the data base
310 PRINT "will be displayed on. (e.g. for the 4051 screen it would be
320 PRINT "72, for a printer it would be 132 characters)"
330 PRINT "The width is: ";
340 INPUT T4
350 DELETE T$
360 DIM T$(T4)
370 PAGE
380 T$="ITEM#"
390 T8=0
400 DIM C(T1,2),W$(133),X$(133),S$(133),P$(133)
410 C(1,1)=5
420 C(1,2)=1
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
430 A$="                                                "
440 A$=A$&A$
450 B$="HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH"
460 B$=B$&B$
470 X$="                                                "
480  $="                                                "
490 X$=REP(C$,1,0)
500 T9=0
510 PRINT "JEnter, in order, width (number of characters) of each ";
520 PRINT "additional field, whether the field will contain numeric";
530 PRINT " (1) or alphanumeric (2) data, and title for the field. ";
540 PRINT "The width of any field can not exceed 70 characters.J"
550 X0=T4-T1-4
560 PRINT "NOTE:The sum of all additional field widths must be     ";
570 PRINT "less than or equal to ";X0;".J"
580 FOR I1=2 TO T1
590     PRINT "JUser Field ";I1;":  Field width (in characters):  ";
600     INPUT C(I1,1)
610     IF C(I1,1)>0 AND C(I1,1)<71 THEN 650
620     PRINT "J* *** Field width is less than one or greater than 70. ";
630     PRINT "Try again *"
640     GO TO 590
650     PRINT "Will entries be numeric (1) or alphanumeric (2):  ";
660     INPUT C(I1,2)
670     IF C(I1,2)<1 OR C(I1,2)>2 THEN 650
680     E$=SEG(A$,1,C(I1,1))
690     F$=SEG(B$,1,C(I1,1))
700     PRINT "Title for field :";E$;F$;
710     INPUT P$
720     T9=T9+C(I1,1)
730     IF T9<=X0 THEN 790
740     PRINT "J* *** Length of record exceeds media width. Try again. *"
750     T$=SEG(T$,1,5)
760     I1=T1
770 NEXT I1
780 GO TO 500
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
790 IF LEN(P$)<=C(I1,1) THEN 840
800 PRINT "J* ***  Title exceeds length of field.  Enter field data ";
810 PRINT "again._*"
820 T9=T9-C(I1,1)
830 GO TO 590
840 IF LEN(P$)=C(I1,1) THEN 870
850 W$=SEG(X$,1,C(I1,1)-LEN(P$))
860 P$=P$&W$
870 T$=REP(P$,1+LEN(T$),0)
880 NEXT I1
890 T8=INT((T4-(T9+5))/(T1-1))
900 S$=SEG(X$,1,T8)
910 S1=T8*(T1-1)+T9+5

920 REM ADD SPACES BETWEEN TITLES

930 I2=6
940 FOR I1=2 TO T1
950     P$=SEG(T$,I2,C(I1,1))
960     P$=S$&P$
970     T$=REP(P$,I2,C(I1,1))
980     I2=I2+C(I1,1)+T8
990 NEXT I1

1000 REM   ***   T2 IS USED FOR THE RECORD COUNTER AND T4 IS THE MEDIA
1010 REM   ***   WIDTH.

1020 T2=0

1030 REM   ***   BEGINING OF THE 4907 DISC FILE INITIALIZATION

1040 CALL "UNIT",A9

1050 REM   ***   REQUEST LIBRARY TITLE FOR DATABASE DEFINITION

1060 DELETE L$,J$
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
1070 J$="Family name"
1080 PRINT "JEnter the database family name: _____HHHHHHHHHH";
1090 INPUT L$
1100 IF L$="" THEN 1080
1110 GOSUB 1570
1120 IF EO THEN 1080
1130 B$="@DATABASE/"&L$
1140 B$=REP("/MASTER",LEN(B$)+1,0)

1150 REM   ***   DETERMINE IF THE NAME GIVEN IS PREVIOUSLY DEFINED

1160 C$=B$&"/INIT"
1170 CALL "FILE",A9,C$,Z$
1180 IF LEN(Z$)=0 THEN 1210
1190 PRINT "J* ***  Files already exist under that family name._*"
1200 GO TO 1080

1210 REM   ***   DEFINE FILEDEF TO CONTAIN RECORD DEFINITION
1220 REM   ***   DEFINE DATABASE TO CONTAIN RECORDS

1230 DIM K$(1)
1240 M$="BU"
1250 PRINT "JDo you want this data base to be a private file? (Y/N): ";
1260 INPUT K$
1270 GO TO POS("DYN",K$,1) OF 1250,1300,1400
1280 GO TO 1250
1290 J$="Password"
1300 PRINT "JEnter a file Password (1 to 10 characters must start with"
1310 PRINT "_a leter but the remaining characters can be alphanumeric.)"
1320 PRINT "Password is: _____HHHHHHHHHH";
1330 INPUT L$
1340 IF L$="" THEN 1300
1350 GOSUB 1570
1360 IF EO THEN 1300
1370 L$=REP(":",1,0)
1380 B$=REP(L$,POS(B$,"/MASTER",9),0)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
1390 M$="BR"
1400 LO=2*T1*10+T4+6 MAX 60
1410 C$=B$&"/INIT"
1420 CREATE C$,"U";LO+0,0
1430 OPEN C$;1,"F",Z$
1440 C$=B$&"/DATA"
1450 CREATE C$,M$;1,T4+5
1460 OPEN C$;2,"F",Z$

1470 REM    ***    ASSIGN LOGICAL UNIT 1 TO FILEDEF
1480 REM    ***    STORE RECORD PARAMETERS

1490 WRITE #1:T1,T2,T9,T8,T4,S1
1500 WRITE #1:C,T$
1510 CLOSE 1
1520 WRITE #2,1:"DATABASE IS EMPTY"
1530 CLOSE
1540 PRINT "J* ***  File initialization complete._*"
1550 PRINT "JJJPress AUTO LOAD key to select other programs."
1560 END

1570 REM
1580 REM      CHECK THE PASSWORD OR FAMILY NAME
1590 REM

1600 EO=0
1610 IF LEN(L$)<11 THEN 1650
1620 PRINT "* *** ";J$;" is longer than ten characters._***  ";
1630 PRINT "Try again._*"
1640 GO TO 1790
1650 P$=SEG(L$,1,1)
1660 K6=ASC(P$)
1670 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 1700
1680 PRINT "* ***  ";J$;" does not start with a letter. Try again._*"
1690 GO TO 1790
1700 FOR I0=1 TO LEN(L$)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
1710      P$=SEG(L$,I0,1)
1720      K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890",P$,1)
1730      K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
1740      IF K6<=0 THEN 1770
1750 NEXT I0
1760 GO TO 1800
1770 PRINT "* ***  ";J$;" contains blanks or non alphanumeric ";
1780 PRINT " characters. ***  Try again. *"
1790 E0=1
1800 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 3

```
1 DELETE 30000,40000
2 SET KEY
3 GO TO 100
4 PRINT @32,26:3
5 RETURN
43 RETURN
44 PRINT @32,26:0
45 RETURN
99 RETURN

100 REM
110 REM ------------------------------------------------------------
120 REM
130 REM                          MAIN   PROGRAM
140 REM
150 REM              GENERALIZED DATA BASE MANAGEMENT PROGRAM
160 REM ***   THIS PROGRAM IS DESIGNED TO OPERATE WITH THE 4907 FLEXABLE
170 REM ***   DISC FILE MANAGER.  THE PROGRAM HAS BEEN SEGMENTED AND
180 REM ***   OVERLAYED TO TAKE ADVANTAGE OF THE DISC'S RAPID ACCESS
190 REM ***   TIME.  THIS ALLOWS LARGE PROGRAMS TO BE RUN WHERE MEMORY IS
200 REM ***   LIMITED.  THE PROGRAM IS TARGETED TO RUN ON A 4051 WITH 8K
210 REM ***   TO 16K OF RAM.
220 REM ***   THIS PROGRAM IS A MODIFIED AND ENHANCED VERSION OF THE ONE
230 REM ***   FOUND IN BP&A VOL. 1  SECTION 15  PP. 4-191 TO 4-212. MUCH
240 REM ***   OF THIS DOCUMENTATION IS NO LONGER APPLICABLE.
250 REM ***                    AUGUST 1978
260 REM
261 REM ***   WARNING:  THIS PROGRAM IS SENSITIVE TO BEING RENUMBERED.
262 REM ***             ANY CHANGES IN THE NUMBERING OF THIS PROGRAM
263 REM ***             MAY REQUIRE CHANGES TO OTHER PROGRAMS.
264 REM

270 H$="@DBMPROG/RECORDKEEP/OV0"
280 GO TO 540

290 REM DIRECTORY
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
300 PRINT "LI              MENU"
310 PRINT "JJI1.  Add items to master database file"
320 PRINT "I2.  Delete items from master database file"
330 PRINT "I3.  Examine or modify an item in master database file"
340 PRINT "JI4.  List local or master database file"
350 PRINT "I5.  Create local database file"
360 PRINT "I6.  Archive local database file"
370 PRINT "JI7.  Transfer archived database to local file"
380 PRINT "I8.  List directory of database files"
390 PRINT "I9.  Delete an archived database file"
400 PRINT "JI10. Sort local database file"
410 PRINT "I11. List local or master database file with subtotals"
420 PRINT "I12. Stop"
430 PRINT "JJJEnter number desired:  ";
440 INPUT A3
450 IF A3<=0 OR A3>12 THEN 290
460 CALL "UNIT",A8
462 DELETE M$,K$
470 K$=STR(A(A3))
480 M$="@DBMPROG/RECORDKEEP/"
490 K$=SEG(K$,2,3)
500 K$=REP("OV",1,0)
510 K$=REP(M$,1,0)
520 IF K$=H$ THEN 570
530 H$=K$

540 REM   ***   THIS STATEMENT IS BRANCHED TO BY OV3 AND OV7.

541 DELETE 4010,8990
550 J2=MEMORY
560 APPEND H$;4000,10
570 CALL "UNIT",A9

580 REM
590 REM ----------------------------------------------------------------
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
600 REM
610 REM                          BEGINNING OF OVERLAY AREA
620 REM
4000 REM      TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM
9000 REM                          END OF OVERLAY AREA
9010 REM
9020 REM ----------------------------------------------------------------
9030 REM

9040 GO TO 290
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 4

```
4000 REM          TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                    OVERLAY     OV1
4050 REM

4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$
4110 H3=1
4120 IF A3=1 THEN 4140
4130 H3=2
4140 GOSUB H3 OF 4170,5350
4150 GO TO 9000

4160 REM
4170 REM ROUTINE TO ADD ITEMS

4180 PRINT "LIROUTINE TO ADD ITEMS"
4190 PRINT "JField titles will appear on the screen in order.  Enter";
4200 PRINT " data for each item following the field title.  To ";
4210 PRINT "discontinue data entry, press RETURN after the first field";
4220 PRINT " title for an item appears.J"

4230 REM   ***   T2 WILL BE THE RECORD COUNTER

4240 L$=""
4250 A$="                                        "
4260 A$=A$&A$
4270 B$="HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH"
4280 B$=B$&B$
4290 GOSUB 6090
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4300 DIM U$(T4),Y$(1)
4310 IF K0=1 THEN 4770
4320 T2=T2+1
4330 U$=""
4340 DELETE O$
4350 DIM O$(5+T8)
4360 PRINT
4370 O$=STR(T2)
4380 I3=5
4390 T5=5+T8+1
4400 GOSUB 4800
4410 IF K0=0 THEN 4430
4420 STOP
4430 U$=REP(O$,1+LEN(U$),0)
4440 R$=""
4450 FOR I4=2 TO T1
4460     DELETE O$,R$,W$
4470     DIM R$(C(I4,1)),W$(C(I4,1)+T8),O$(C(I4,1)+T8)
4480     R$=SEG(T$,T5,C(I4,1))
4490     C$=SEG(A$,1,C(I4,1))
4500     D$=SEG(B$,1,C(I4,1))
4510     PRINT "Enter ";R$;"_ :";C$;D$;
4520     INPUT O$
4530     IF O$="" AND I4<>2 THEN 4510
4540     IF O$<>"" OR I4<>2 THEN 4580

4550     REM   ***   EXIT FROM THE ENTRY PORTION

4560     T2=T2-1
4570     GO TO 4730
4580     IF LEN(O$)<=C(I4,1) THEN 4620
4590     PRINT "J* *** Length of data exceeds field of ";C(I4,1);
4600     PRINT " characters._*** Try again._*"
4610     GO TO 4510
4620     I3=C(I4,1)
4630     GOSUB C(I4,2) OF 4800,5310
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4640      IF K0=0 THEN 4660
4650      GO TO 4510
4660      O$=S$&O$
4670      U$=REP(O$,1+LEN(U$),0)
4680      T5=T5+C(I4,1)+T8
4690 NEXT I4

4700 REM    ***    WRITE NEW RECORD TO DISC FILE

4710 WRITE #2,T2:U$
4720 GO TO 4320

4730 REM    ***    UPDATE RECORD DEFINITION INFO T2=NUMBER OF RECORDS

4740 CALL "REWIND",1
4750 WRITE #1:T1,T2,T9,T8,T4,S1
4760 WRITE #1:C,T$
4770 CLOSE
4780 DELETE A$,B$,C$,D$
4790 RETURN

4800 REM    VERIFY THE STRING IS NUMERIC THEN RIGHT JUSTIFY

4810 RESTORE 4820
4820 DATA 0,0,0,0,0,0
4830 READ K0,H4,K1,K2,K3,K4
4840 DELETE J$,V$
4850 H1=LEN(O$)
4860 DIM J$(H1),V$(1)
4870 J$=""
4880 V$=""
4890 FOR H2=1 TO H1
4900      Y$=SEG(O$,H2,1)
4910      K4=POS("0123456789.+-E, ",Y$,1)
4920      IF K4<=0 THEN 5280
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4930     REM   CASE 1 -  Y$= A COMMA OR BLANK   "IGNORE IT"

4940     IF K4=15 OR K4=16 THEN 5040
4950     IF K4=14 THEN 5120
4960     IF K4=12 OR K4=13 THEN 5190
4970     IF K4=11 THEN 5230

4980     REM   Y$ IS A NUMERIC CHARACTER

4990     K3=1

5000     REM   ADD CHARACTER TO OUTPUT STRING (J$) AND INC. COUNTER (H4)

5010     H4=H4+1
5020     J$=REP(Y$,LEN(J$)+1,0)
5030     V$=Y$
5040 NEXT H2

5050 REM   CHECK J$ FOR ZERO LENGHT OR NO DIGITS

5060 IF K3=0 OR LEN(J$)<=0 THEN 5280

5070 REM   PACK NUMERIC STRING RIGHT JUSTIFIED

5080 O$=J$
5090 W$=SEG(X$,1,I3-LEN(O$))
5100 O$=REP(W$,1,0)
5110 RETURN

5120 REM   CASE 2 -  Y$= "E"

5130 IF LEN(V$)=0 THEN 5280
5140 IF ASC(V$)=43 OR ASC(V$)=45 THEN 5280
5150 K1=1+K1
5160 IF K1>1 THEN 5280
5170 K3=0
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5180 GO TO 5000

5190 REM   CASE 3 -  Y$= A PLUS OR MINUS SIGN

5200 IF V$="" THEN 5000
5210 IF ASC(V$)<>69 THEN 5280
5220 GO TO 5000

5230 REM   CASE 4 -  Y$= A DECIMAL POINT

5240 K2=K2+1
5250 IF K2>1 THEN 5280
5260 IF K1=>1 THEN 5280
5270 GO TO 5000
5280 PRINT "J* *** Illegal numeric entry. Try again. *"
5290 K0=1
5300 RETURN

5310 REM PACK W/SPACES-ALPHANUMERIC

5320 W$=SEG(X$,1,I3-LEN(O$))
5330 O$=O$&W$
5340 RETURN

5350 REM ROUTINE TO DELETE AN ITEM

5360 H9=0
5370 L$=""
5380 GOSUB 6090
5390 DIM U$(T4),Y$(1)
5400 IF K0=1 THEN 5450
5410 READ #2,1:I$
5420 IF I$<>"DATABASE IS EMPTY" THEN 5470
5430 PRINT "J* *** The MASTER database is EMPTY. Delete command ";
5440 PRINT "aborted. *"
5450 PRINT USING "4(T10(""H"")),"" KK""":
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5460 GO TO 6070
5470 CALL "REWIND",2
5480 U$=""
5490 PRINT "JEnter Item No. to be deleted: ";
5500 INPUT L$
5510 IF L$="" THEN 5490

5520 REM    CHECK FOR NON-NUMERIC CHARACTERS

5530 FOR K5=1 TO LEN(L$)
5540     J$=SEG(L$,K5,1)
5550     IF J$=>"0" AND J$<="9" OR J$=" " THEN 5590
5560     PRINT "J* ***  Previous data entered contained non-numeric data."
5570     PRINT "***  Try again._*"
5580     GO TO 5490
5590 NEXT K5
5600 H8=VAL(L$)
5610 IF H8<=0 THEN 5830
5620 IF H8>0 AND H8<=T2 THEN 5650
5630 PRINT "J* ***  Item number must fall between 1 and ";T2;"_*"
5640 GO TO 5490

5650 REM   ***   READ SELECTED RECORD FROM DISC FILE

5660 READ #2,H8:U$
5670 GOSUB 6420
5680 PRINT "L";T$;"J"
5690 PRINT U$
5700 GOSUB 6470
5710 PRINT "JIs this the item you wish to delete (Y/N):  ";
5720 INPUT Y$
5730 IF Y$="Y" THEN 5760
5740 IF Y$<>"N" THEN 5710
5750 GO TO 5470
5760 U$="~~~~~"
5770 H9=1
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5780 REM   ***   FLAG RECORD TO BE REMOVED DURING PACK

5790 WRITE #2,H8:U$
5800 PRINT "JDo you wish to delete another item? (Y/N): ";
5810 INPUT Y$
5820 IF Y$="Y" THEN 5470

5830 REM   ***   PACK DELETED RECORDS FROM DATABASE

5840 IF H9=0 THEN 6070
5850 I0=1
5860 FOR I1=1 TO T2
5870     READ #2,I1:U$
5880     I$=SEG(U$,1,5)
5890     IF I$="~~~~~" THEN 5990
5900     IF I0=I1 THEN 5980
5910     I3=5
5920     O$=STR(I0)
5930     GOSUB 4800
5940     IF K0=0 THEN 5960
5950     STOP
5960     U$=REP(O$,1,5)
5970     WRITE #2,I0:U$
5980     I0=I0+1
5990 NEXT I1
6000 I0=I0-1
6010 T2=I0
6020 IF T2>0 THEN 6040
6030 WRITE #2,1:"DATABASE IS EMPTY"
6040 CALL "REWIND",1
6050 WRITE #1:T1,T2,T9,T8,T4,S1
6060 WRITE #1:C,T$
6070 CLOSE
6080 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
6090 REM    ***    OPEN DATABASE FILES

6100 CLOSE
6110 K0=0
6120 GOSUB 6270
6130 CALL "FILE",A9,E$,K$
6140 IF LEN(K$)>0 THEN 6190
6150 K0=1
6160 PRINT "J* ***  File was not found on disc._*"
6170 PRINT USING "4(110(""H"")),"" KK""":
6180 GO TO 6260
6190 OPEN E$;1,"F",K$
6200 OPEN F$;2,"F",K$
6210 READ #1:T1,T2,T9,T8,T4,S1
6220 DELETE C,T$
6230 DIM C(T1,2),T$(T4)
6240 READ #1:C,T$
6250 S$=SEG(X$,1,T8)
6260 RETURN

6270 REM     SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
6280 REM    L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

6290 K$="/LOCAL/INIT/"
6300 K$=REP(L$,LEN(K$)+1,0)
6310 IF L$<>"" THEN 6340
6320 K$=REP("MASTER",2,5)
6330 K$=SEG(K$,1,LEN(K$)-1)
6340 E$=G$&K$
6350 K$="/LOCAL/DATA/"
6360 K$=REP(L$,LEN(K$)+1,0)
6370 IF L$<>"" THEN 6400
6380 K$=REP("MASTER",2,5)
6390 K$=SEG(K$,1,LEN(K$)-1)
6400 F$=G$&K$
6410 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
6420 REM    SUBROUTINE TO SELECT CHARACTER SIZE ON 4054

6430 IF RND(0)<0.8 THEN 6460
6440 K9=1+(T4-73<0)+(T4-80<0)+(T4-120<0)
6450 PRINT @32,17:K9
6460 RETURN

6470 REM     SUBROUTINE TO RESET CHARACTER SIZE ON 4054

6480 IF RND(0)<0.8 THEN 6500
6490 PRINT @32,17:4
6500 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 5

```
4000 REM          TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                    OVERLAY    OV2
4050 REM

4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$
4110 GOSUB 4140
4120 GO TO 9000

4130 REM
4140 REM    ROUTINE TO EXAMINE OR MODIFY

4150 L$=""
4160 A$="                                        "
4170 A$=A$&A$
4180 B$="HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH"
4190 B$=B$&B$
4200 GOSUB 4900
4210 DIM I$(T4),U$(T4),Y$(1)
4220 IF K0 THEN 4270
4230 READ #2,1:I$
4240 IF I$<>"DATABASE IS EMPTY" THEN 4290
4250 PRINT "J* ***  The MASTER database is EMPTY. Modify command ";
4260 PRINT "aborted. *"
4270 PRINT USING "4(T10(""H"")),"" KK""":
4280 GO TO 4870
4290 CALL "REWIND",2
4300 O$=""
4310 I2=0
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4320 PRINT "JEnter Item No. to be examined:  ";
4330 INPUT L$
4340 IF L$="" THEN 4320

4350 REM    CHECK FOR NON-NUMERIC CHARACTERS

4360 FOR K5=1 TO LEN(L$)
4370     J$=SEG(L$,K5,1)
4380     IF J$=>"0" AND J$<="9" OR J$=" " THEN 4420
4390     PRINT "J* ***  Previous data entered contained non-numeric data."
4400     PRINT "***  Try again._*"
4410     GO TO 4320
4420 NEXT K5
4430 I1=VAL(L$)
4440 IF I1<=0 THEN 4870
4450    T1>0 AND I1<=T2 THEN 4480
4460 PRINT "J* ***  Item number must fall between 1 and ";T2;"_*"
4470 GO TO 4320

4480 REM   ***   READ SELECTED RECORD FROM DISC FILE

4490 GOSUB 5780
4500 PRINT "L";T$;"J"
4510 READ #2,I1:U$
4520 PRINT U$
4530 GOSUB 5830
4540 PRINT "JJDo you wish to modify this item (Y/N):  ";
4550 INPUT Y$
4560 IF Y$="N" THEN 4830
4570 IF Y$<>"Y" THEN 4540
4580 PRINT "JTo leave a data item as it is, press RETURN."
4590 PRINT "To modify, enter correct data."
4600 T6=6+T8
4610 FOR I4=2 TO T1
4620     DELETE P$,R$,O$
4630     DIM P$(C(I4,1)),R$(C(I4,1))
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4640      P$=SEG(T$,T6,C(I4,1))
4650      R$=SEG(U$,T6,C(I4,1))
4660      C$=SEG(A$,1,C(I4,1))
4670      D$=SEG(B$,1,C(I4,1))
4680      PRINT "_";P$;"_OLD: ";R$;"_NEW_ :";C$;D$;
4690      INPUT O$
4700      IF O$="" THEN 4790
4710      IF LEN(O$)<=C(I4,1) THEN 4750
4720      PRINT "J* ***  Length of data exceeds field of ";C(I4,1);
4730      PRINT "_characters._***  Try again._*"
4740      GO TO 4680
4750      I3=C(I4,1)
4760      GOSUB C(I4,2) OF 5230,5740
4770      IF K0=1 THEN 4680
4780      U$=REP(O$,T6,C(I4,1))
4790      T6=T6+C(I4,1)+T8
4800 NEXT I4

4810 REM   ***   REPLACE SELECTED RECORD IN DISC FILE

4820 WRITE #2,I1:U$
4830 PRINT "JDo you want to examine another item (Y/N):  ";
4840 INPUT Y$
4850 IF Y$="Y" THEN 4300
4860 IF Y$<>"N" THEN 4830
4870 CLOSE
4880 DELETE A$,B$,C$,D$
4890 RETURN

4900 REM   ***   OPEN DATABASE FILES

4910 CLOSE
4920 K0=0
4930 GOSUB 5080
4940 CALL "FILE",A9,E$,K$
4950 IF LEN(K$)>0 THEN 5000
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4960 K0=1
4970 PRINT "J* ***  File was not found on disc. *"
4980 PRINT USING "4(110(""H"")),"" KK""":
4990 GO TO 5070
5000 OPEN E$;1,"F",K$
5010 OPEN F$;2,"F",K$
5020 READ #1:T1,T2,T9,T8,T4,S1
5030 DELETE C,T$
5040 DIM C(T1,2),T$(T4)
5050 READ #1:C,T$
5060 S$=SEG(X$,1,T8)
5070 RETURN

5080 REM    SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
5090 REM   L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

5100 K$="/LOCAL/INIT/"
5110 K$=REP(L$,LEN(K$)+1,0)
5120 IF L$<>"" THEN 5150
5130 K$=REP("MASTER",2,5)
5140 K$=SEG(K$,1,LEN(K$)-1)
5150 E$=G$&K$
5160 K$="/LOCAL/DATA/"
5170 K$=REP(L$,LEN(K$)+1,0)
5180 IF L$<>"" THEN 5210
5190 K$=REP("MASTER",2,5)
5200 K$=SEG(K$,1,LEN(K$)-1)
5210 F$=G$&K$
5220 RETURN

5230 REM    VERIFY THE STRING IS NUMERIC THEN RIGHT JUSTIFY

5240 RESTORE 5250
5250 DATA 0,0,0,0,0,0
5260 READ K0,H4,K1,K2,K3,K4
5270 DELETE J$,V$
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5280 H1=LEN(O$)
5290 DIM J$(H1),V$(1)
5300 J$=""
5310 V$=""
5320 FOR H2=1 TO H1
5330      Y$=SEG(O$,H2,1)
5340      K4=POS("0123456789.+-E, ",Y$,1)
5350      IF K4<=0 THEN 5710

5360      REM   CASE 1 -  Y$= A COMMA OR BLANK   "IGNORE IT"

5370      IF K4=15 OR K4=16 THEN 5470
5380      IF K4=14 THEN 5550
5390      IF K4=12 OR K4=13 THEN 5620
5400      IF K4=11 THEN 5660

5410      REM   Y$ IS A NUMERIC CHARACTER

5420      K3=1

5430      REM   ADD CHARACTER TO OUTPUT STRING (J$) AND INC. COUNTER (H4)

5440      H4=H4+1
5450      J$=REP(Y$,LEN(J$)+1,0)
5460      V$=Y$
5470 NEXT H2

5480 REM   CHECK J$ FOR ZERO LENGHT OR NO DIGITS

5490 IF K3=0 OR LEN(J$)<=0 THEN 5710

5500 REM   PACK NUMERIC STRING RIGHT JUSTIFIED

5510 O$=J$
5520 W$=SEG(X$,1,I3-LEN(O$))
5530 O$=REP(W$,1,0)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5540 RETURN

5550 REM   CASE 2 -  Y$= "E"

5560 IF LEN(V$)=0 THEN 5710
5570 IF ASC(V$)=43 OR ASC(V$)=45 THEN 5710
5580 K1=1+K1
5590 IF K1>1 THEN 5710
5600 K3=0
5610 GO TO 5430

5620 REM   CASE 3 -  Y$= A PLUS OR MINUS SIGN

5630 IF V$="" THEN 5430
5640 IF ASC(V$)<>69 THEN 5710
5650 GO TO 5430

5660 REM   CASE 4 -  Y$= A DECIMAL POINT

5670 K2=K2+1
5680 IF K2>1 THEN 5710
5690 IF K1=>1 THEN 5710
5700 GO TO 5430
5710 PRINT "J*_***  Illegal numeric entry. Try again._*"
5720 K0=1
5730 RETURN

5740 REM PACK W/SPACES-ALPHANUMERIC

5750 W$=SEG(X$,1,I3-LEN(O$))
5760 O$=O$&W$
5770 RETURN

5780 REM    SUBROUTINE TO SELECT CHARACTER SIZE ON 4054

5790 IF RND(0)<0.8 THEN 5820
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5800 K9=1+(T4-73<0)+(T4-80<0)+(T4-120<0)
5810 PRINT @32,17:K9
5820 RETURN

5830 REM      SUBROUTINE TO RESET CHARACTER SIZE ON 4054

5840 IF RND(0)<0.8 THEN 5860
5850 PRINT @32,17:4
5860 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 6

```
4000 REM           TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                        OVERLAY     OV3
4050 REM

4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$

4110 REM CREATE A LOCAL DATABASE
4120 REM    OPEN THE SOURCE DATABASE

4130 PRINT "JEnter the name of the LOCAL database (i.e. Source data";
4140 PRINT "base) from which you intend to create the new local data";
4150 PRINT "base, or enter only a CR if you wish to use the MASTER ";
4160 PRINT "database.          HHHHHHHHHH";
4170 INPUT L$
4180 IF L$="!M" THEN 5510
4190 IF L$="" THEN 4220
4200 GOSUB 5870
4210 IF E0 THEN 4130
4220 R$=L$
4230 GOSUB 5540
4240 IF E0 THEN 5510
4250 IF R$<>"" THEN 4270
4260 L$="MASTER"
4270 READ #2,1:I$
4280 IF I$<>"DATABASE IS EMPTY" THEN 4310
4290 PRINT "J* ***  ";L$;" is EMPTY. Try again._*"
4300 GO TO 4120
4310 T0=0
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4320 IF T2>0 THEN 4370
4330 PRINT "J* ***  The MASTER database is EMPTY. Create command ";
4340 PRINT "aborted. *"
4350 PRINT USING "4(110(""H"")),"" KK""":
4360 GO TO 5510
4370 PRINT "LThe next step is to assign database fields to variables ";
4380 PRINT "which will be used in the following logical expression.";
4390 PRINT " This is done in two parts, and each can be terminated ";
4400 PRINT "by responding with a CR to a prompt. Any numeric field may";
4410 PRINT "be assigned to one of the variables F1,F2,F3, or F4, and ";
4420 PRINT "similarly any field may be assigned to one of the chara";
4430 PRINT "cter variables A$,B$,C$, or D$."
4440 DELETE A$,B$,C$,D$,F1,F2,F3,F4,H0
4450 RESTORE 4460
4460 DATA 0,0,0,0," "," "," "," "
4470 READ F1,F2,F3,F4,A$,B$,C$,D$
4480 DIM H0(4,2)

4490 REM * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
4500 REM *
4510 REM ***     CONSTRUCT H0 MATRIX.  H0 SPECIFIES WHICH ITEM FIELD
4520 REM ***     IS ASSIGNED TO WHICH DATA VARIABLES.
4530 REM *

4540 H0=-100
4550 M$="F1F2F3F4A$B$C$D$"
4560 N$="O$P$Q$R$"
4570 PRINT "JNumeric variables will be assigned first. Choose from ";
4580 PRINT "list below.J"
4590 H7=1
4600 FOR I5=1 TO T1
4610     IF C(I5,2)=2 THEN 4660
4620     DELETE I$
4630     DIM I$(C(I5,1))
4640     I$=SEG(T$,H7,C(I5,1))
4650     PRINT USING "4X,3D,"")"",2X,62A":I5,I$
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4660     H7=H7+C(I5,1)+T8
4670 NEXT I5
4680 PRINT " "
4690 I9=100

4700 REM         OPEN FILE THAT PROGRAM STEPS WILL BE WRITTEN TO

4710 CALL "UNIT",A8
4720 K$="@DBMPROG/LOGIC"
4730 CALL "FILE",A8,K$,L$
4740 IF LEN(L$)>0 THEN 4760
4750 CREATE K$,"AU";100,0
4760 OPEN "@DBMPROG/LOGIC";8,"F",K$
4770 FOR J0=1 TO 2
4780     FOR K0=1 TO 4
4790         K$=SEG(M$,((J0-1)*4+(K0-1))*2+1,2)
4800         PRINT "Enter the field number you wish to assign to variable ";
4810         PRINT K$;": ";
4820         INPUT L$
4830         IF L$="" THEN 5330

4840         REM    CHECK FOR NON NUMERIC CHARACTERS

4850         FOR K5=1 TO LEN(L$)
4860             J$=SEG(L$,K5,1)
4870             IF J$=>"0" AND J$<="9" OR J$=" " THEN 4910
4880             PRINT "J* ***  Previous data entered contained non-numeric d
4890             PRINT "***  Try again._*"
4900             GO TO 4800
4910         NEXT K5
4920         I0=VAL(L$)
4930         IF I0=0 THEN 5510
4940         IF I0>0 AND I0<=T1 THEN 4970
4950         PRINT "J* ***  The field number is not between 1 and ";T1;"._*"
4960         GO TO 4800
4970         IF J0=2 THEN 5010
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4980        IF C(I0,2)=1 THEN 5070
4990        PRINT "J* ***  Field #";I0;" is not a numeric field._*"
5000        GO TO 4800
5010        IF C(I0,2)=2 THEN 5070
5020        PRINT "J* ***  Field #";I0;" is not an alphanumeric field._*"
5030        PRINT "JDo you want it treated as one? (Y/N): ";
5040        INPUT Y$
5050        GO TO POS("YN",Y$,1) OF 5070,4800
5060        GO TO 5020
5070        H0(K0,J0)=I0

5080        REM     STORE BEGINNING AND LENGTH OF FIELD IN V$ AND W$

5090        H7=1
5100        FOR I1=1 TO T1
5110            IF I1=I0 THEN 5150
5120            H7=H7+T8+C(I1,1)
5130        NEXT I1
5140        STOP
5150        V$=STR(H7)
5160        W$=STR(C(I0,1))
5170        IF J0=2 THEN 5250

5180        REM   WRITE PROGRAM STEPS FOR NUMERIC VARIABLES

5190        O$=SEG(N$,(K0-1)*2+1,2)
5200        I9=I9+10
5210        PRINT #8:I9;" ";O$;"=SEG(U$,";V$;",";W$;") "
5220        I9=I9+10
5230        PRINT #8:I9;" ";K$;"=VAL(";O$;") "
5240        GO TO 5340

5250        REM   WRITE PROGRAM STEPS FOR ALPHANUMERIC VARIABLES

5260        I9=I9+10
5270        PRINT #8:I9;" V$=SEG(U$,";V$;",";W$;") "
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5280          I9=I9+10
5290          PRINT #8:I9;" GOSUB 4090 "
5300          I9=I9+10
5310          PRINT #8:I9;" ";K$;"=V$ "
5320          GO TO 5340
5330          K0=4
5340      NEXT K0
5350      IF J0=2 THEN 5460
5360      PRINT "LAlphanumeric variables will now be assigned.J"
5370      H7=1
5380      FOR I5=1 TO T1
5390          DELETE I$
5400          DIM I$(C(I5,1))
5410          I$=SEG(T$,H7,C(I5,1))
5420          PRINT USING "4X,3D,"")"",2X,62A":I5,I$
5430          H7=H7+C(I5,1)+T8
5440      NEXT I5
5450      PRINT " "
5460 NEXT J0

5470 REM  CALL OVERLAY  OV3A

5480 CALL "UNIT",A8
5490 H$="@DBMPROG/RECORDKEEP/OV3A"
5500 GO TO 540

5510 REM  ABORT CREATION OF LOCAL FILE

5520 CLOSE
5530 GO TO 9000

5540 REM   ***   OPEN DATABASE FILES

5550 CLOSE
5560 E0=0
5570 GOSUB 5720
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5580 CALL "FILE",A9,E$,K$
5590 IF LEN(K$)>0 THEN 5640
5600 E0=1
5610 PRINT "J* ***  File was not found on disc._*"
5620 PRINT USING "4(110(""H"")),"" KK""":
5630 GO TO 5710
5640 OPEN E$;1,"F",K$
5650 OPEN F$;2,"F",K$
5660 READ #1:T1,T2,T9,T8,T4,S1
5670 DELETE C,T$
5680 DIM C(T1,2),T$(T4)
5690 READ #1:C,T$
5700 S$=SEG(X$,1,T8)
5710 RETURN

5720 REM     SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
5730 REM    L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

5740 K$="/LOCAL/INIT/"
5750 K$=REP(L$,LEN(K$)+1,0)
5760 IF L$<>"" THEN 5790
5770 K$=REP("MASTER",2,5)
5780 K$=SEG(K$,1,LEN(K$)-1)
5790 E$=G$&K$
5800 K$="/LOCAL/DATA/"
5810 K$=REP(L$,LEN(K$)+1,0)
5820 IF L$<>"" THEN 5850
5830 K$=REP("MASTER",2,5)
5840 K$=SEG(K$,1,LEN(K$)-1)
5850 F$=G$&K$
5860 RETURN

5870 REM
5880 REM          CHECK THE PASSWORD OR FILE NAME IN L$
5890 REM
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5900 E0=0
5910 IF LEN(L$)<11 THEN 5940
5920 PRINT "* ***  The name is longer than ten characters. Try again._*"
5930 GO TO 6080
5940 P$=SEG(L$,1,1)
5950 K6=ASC(P$)
5960 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 5990
5970 PRINT "* ***  The name does not start with a letter. Try again._*"
5980 GO TO 6080
5990 FOR I0=1 TO LEN(L$)
6000     P$=SEG(L$,I0,1)
6010     K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
6020     K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
6030     IF K6<=0 THEN 6060
6040 NEXT I0
6050 GO TO 6090
6060 PRINT "* ***  The name contains blanks or non alphanumeric ";
6070 PRINT "characters._***  Try again._*"
6080 E0=1
6090 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 7

```
4000 REM           TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                          OVERLAY     OV3A
4050 REM

4060 DELETE 5210,5420
4070 K1=MEMORY
4080 GO TO 4320

4090 REM     SUBROUTINE TO STRIP TRAILING BLANKS FROM V$
4100 REM

4110 FOR I8=LEN(V$) TO 1 STEP -1
4120      Y$=SEG(V$,I8,1)
4130      IF Y$<>" " THEN 4150
4140 NEXT I8
4150 V$=SEG(V$,1,I8)
4160 RETURN

4170 REM     SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
4180 REM    L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

4190 K$="/LOCAL/INIT/"
4200 K$=REP(L$,LEN(K$)+1,0)
4210 IF L$<>"" THEN 4240
4220 K$=REP("MASTER",2,5)
4230 K$=SEG(K$,1,LEN(K$)-1)
4240 E$=G$&K$
4250 K$="/LOCAL/DATA/"
4260 K$=REP(L$,LEN(K$)+1,0)
4270 IF L$<>"" THEN 4300
4280 K$=REP("MASTER",2,5)
4290 K$=SEG(K$,1,LEN(K$)-1)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4300 F$=G$&K$
4310 RETURN

4320 REM ***     CONSTRUCT THE USER SUPPLIED LOGICAL EXPRESSION

4330 DELETE L,L$
4340 PRINT "LAt this point you must enter a logical expression. When ";
4350 PRINT "this expression is true for an item in the MASTER database";
4360 PRINT " it will be included in the LOCAL database being created. ";
4370 PRINT "The only constraints on the expression are: 1) only data ";
4380 PRINT "variables assigned to fields above can be used, 2) the ";
4390 PRINT "expression must not have syntax errors, and 3) it can be ";
4400 PRINT "no longer than 65 characters. Fields are listed below.J"
4410 FOR I4=1 TO 2
4420     FOR I5=1 TO 4
4430         K$=SEG(M$,((I4-1)*4+(I5-1))*2+1,2)
4440         H7=1
4450         IF H0(I5,I4)<0 THEN 4540
4460         FOR I6=1 TO T1
4470             IF H0(I5,I4)<>I6 THEN 4510
4480             L$=SEG(T$,H7,C(I6,1))
4490             PRINT USING "4X,2A,"" = """,63A":K$;L$
4500             I6=T1
4510             H7=H7+C(I6,1)+T8
4520         NEXT I6
4530         GO TO 4550
4540         I5=4
4550     NEXT I5
4560     IF I4=2 THEN 4580
4570     PRINT " "
4580 NEXT I4
4590 DELETE L$
4600 DIM L$(65)
4610 PRINT "JPlease enter the logical expression below. J     IF ";
4620 PRINT "_____________________________________________________________";
4630 PRINT "_________K_________";
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4640 INPUT L$
4650 IF L$="" THEN 4340
4660 I9=I9+10
4670 PRINT #8:I9;" L=";L$
4680 PRINT #8:""
4690 CLOSE 8

4700 REM   CREATE DISC FILE FOR LOCAL DATABASE

4710 PRINT "JEnter the name (1 to 10 characters) of the LOCAL ";
4720 PRINT "database: __________HHHHHHHHHH";
4730 INPUT L$
4740 IF L$="" THEN 4710
4750 IF L$="!M" THEN 5570
4760 GOSUB 5600
4770 IF E0 THEN 4710
4780 IF R$="" OR R$<>L$ THEN 4820
4790 PRINT "J* ***  ";R$;" was used above for the source database."
4800 PRINT "***  Choose another name._*"
4810 GO TO 4710
4820 GOSUB 4170
4830 CALL "FILE",A9,E$,K$
4840 IF LEN(K$)=0 THEN 4920
4850 PRINT "J* ***  Local file name in use._*"
4860 PRINT "JDo you want to overwrite the Local file (Y/N): ";
4870 INPUT Y$
4880 GO TO POS("YN",Y$,1) OF 4900,4710
4890 GO TO 4860
4900 KILL E$
4910 KILL F$
4920 L0=(2*T1+5)*10+T4+3
4930 CREATE E$,"U";L0+0,0
4940 CREATE F$,"U";1,T4+5
4950 OPEN E$;8,"F",M$
4960 OPEN F$;7,"F",M$
4970 WRITE #8:T1,0,T9,T8,T4,S1
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4980 WRITE #8:C,T$
4990 WRITE #7,1:"DATABASE IS EMPTY"
5000 CALL "REWIND",8
5010 CALL "REWIND",7
5020 CALL "UNIT",A8
5030 K1=MEMORY
5040 APPEND "@DBMPROG/LOGIC","ASCII";5200
5050 CALL "UNIT",A9
5060 DELETE A$,B$,C$,D$,O$,P$,Q$,R$,V$
5070 DIM A$(T4),B$(T4),C$(T4),D$(T4),P$(T4),Q$(T4),R$(T4),O$(T4),V$(T4

5080 REM *
5090 REM ***     WRITE LOCAL DATABASE
5100 REM *

5110 T0=0
5120 DELETE U$
5130 DIM U$(T4)
5140 FOR J3=1 TO T2
5150      READ #2,J3:U$

5160      REM
5170      REM ----------------------------------------------------------
5180      REM
5190      REM          PROGRAM STEPS WRITTEN TO DISC BY OV3 APPENDED HERE
5200      REM     TARGET LINE FOR PROGRAM STEP APPEND
5210      REM
5220      REM
5230      REM
5240      REM
5250      REM
5260      REM
5270      REM
5280      REM
5290      REM
5300      REM
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5310      REM
5320      REM
5330      REM
5340      REM
5350      REM
5360      REM
5370      REM
5380      REM
5390      REM
5400      REM
5410      REM
5420      REM
5430      REM
5440      REM ------------------------------------------------------------
5450      REM
5460      REM                      END OF APPENDED PROGRAM STEPS
5470      REM

5480      IF NOT(L) THEN 5510
5490      T0=T0+1
5500      WRITE #7,T0:U$
5510 NEXT J3
5520 IF T0>0 THEN 5550
5530 PRINT "J* ***  WARNING: Created database is EMPTY!_*"
5540 PRINT USING "5(110(2("" H"")))," KK""":
5550 WRITE #8:T1,T0,T9,T8,T4,S1
5560 WRITE #8:C,T$
5570 CLOSE
5580 DELETE A$,B$,C$,D$
5590 GO TO 9000

5600 REM
5610 REM         CHECK THE PASSWORD OR FILE NAME IN L$
5620 REM

5630 E0=0
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5640 IF LEN(L$)<11 THEN 5670
5650 PRINT "* ***  The name is longer than ten characters. Try again._*"
5660 GO TO 5810
5670 P$=SEG(L$,1,1)
5680 K6=ASC(P$)
5690 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 5720
5700 PRINT "* ***  The name does not start with a letter. Try again._*"
5710 GO TO 5810
5720 FOR I0=1 TO LEN(L$)
5730     P$=SEG(L$,I0,1)
5740     K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
5750     K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
5760     IF K6<=0 THEN 5790
5770 NEXT I0
5780 GO TO 5820
5790 PRINT "* ***  The name contains blanks or non alphanumeric ";
5800 PRINT "characters._***  Try again._*"
5810 I0=1
5820 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 8

```
4000 REM           TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                          OVERLAY     OV4
4050 REM

4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$
4110 H3=1
4120 IF A3=6 THEN 4140
4130 H3=2
4140 GOSUB H3 OF 4170,4630
4150 GO TO 9000

4160 REM
4170 REM    ARCHIVE A LOCAL DATABASE

4180 CLOSE
4190 PRINT "JEnter the LOCAL database name you wish to archive: ";
4200 PRINT "          HHHHHHHHHH";
4210 INPUT L$
4220 IF L$="" THEN 4190
4230 IF L$="!M" THEN 4610
4240 GOSUB 5170
4250 IF E0 THEN 4190
4260 GOSUB 5020
4270 M$=E$
4280 N$=F$
4290 CALL "FILE",A9,M$,K$
4300 IF LEN(K$)>0 THEN 4330
4310 PRINT "J* *** LOCAL database was not found on disc. *"
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4320 GO TO 4190
4330 OPEN M$;1,"R",Q$
4340 READ #1:T1,T2,T9,T8,T4,S1
4350 CLOSE
4360 IF T2>0 THEN 4390
4370 PRINT "J* ***  The local database is empty._*"
4380 GO TO 4190
4390 PRINT "JEnter the file name for the ARCHIVE copy, or if it is the";
4400 PRINT " same as the local file name enter only a carriage return."
4410 PRINT "Enter name: _____HHHHHHHHHH";
4420 INPUT L$
4430 IF L$="!M" THEN 4610
4440 IF L$="" THEN 4480
4450 GOSUB 5170
4460 IF E0 THEN 4390
4470 GOSUB 5020
4480 E$=REP("ARCHIVE",LEN(G$)+2,5)
4490 F$=REP("ARCHIVE",LEN(G$)+2,5)
4500 CALL "FILE",A9,E$,K$
4510 IF LEN(K$)=0 THEN 4590
4520 PRINT "J* ***  Archive file name in use._*"
4530 PRINT "JDo you want to overwrite the Archive file (Y/N): ";
4540 INPUT Y$
4550 GO TO POS("YN",Y$,1) OF 4570,4390
4560 GO TO 4530
4570 KILL E$
4580 KILL F$
4590 COPY M$,A9 TO E$,A9
4600 COPY N$,A9 TO F$,A9
4610 CLOSE
4620 RETURN

4630 REM    COPY AN ARCHIVE FILE TO LOCAL FILE

4640 PRINT "JEnter the name of the ARCHIVED database to become a local ";
4650 PRINT " file._Enter name: _____HHHHHHHHHH";
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4660 INPUT L$
4670 IF L$="" THEN 4640
4680 IF L$="!M" THEN 5000
4690 GOSUB 5170
4700 IF EO THEN 4640
4710 GOSUB 5020
4720 M$=E$
4730 N$=F$
4740 M$=REP("ARCHIVE",LEN(G$)+2,5)
4750 N$=REP("ARCHIVE",LEN(G$)+2,5)
4760 CALL "FILE",A9,M$,K$
4770 IF LEN(K$)>0 THEN 4800
4780 PRINT "J* *** Archived database was not found on disc. *"
4790 GO TO 4640
4800 PRINT "JEnter the LOCAL file name, or only a carriage return if ";
4810 PRINT "you want the local file name the same as the archived file";
4820 PRINT ". Enter name: _____HHHHHHHHHH";
4830 INPUT L$
4840 IF L$="!M" THEN 5000
4850 IF L$="" THEN 4890
4860 GOSUB 5170
4870 IF EO THEN 4800
4880 GOSUB 5020
4890 CALL "FILE",A9,E$,K$
4900 IF LEN(K$)=0 THEN 4980
4910 PRINT "J* *** Local file name in use. *"
4920 PRINT "JDo you want to overwrite the Local file (Y/N): ";
4930 INPUT Y$
4940 GO TO POS("YN",Y$,1) OF 4960,4800
4950 GO TO 4920
4960 KILL E$
4970 KILL F$
4980 COPY M$,A9 TO E$,A9
4990 COPY N$,A9 TO F$,A9
5000 CLOSE
5010 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5020 REM     SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
5030 REM    L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

5040 K$="/LOCAL/INIT/"
5050 K$=REP(L$,LEN(K$)+1,0)
5060 IF L$<>"" THEN 5090
5070 K$=REP("MASTER",2,5)
5080 K$=SEG(K$,1,LEN(K$)-1)
5090 E$=G$&K$
5100 K$="/LOCAL/DATA/"
5110 K$=REP(L$,LEN(K$)+1,0)
5120 IF L$<>"" THEN 5150
5130 K$=REP("MASTER",2,5)
5140 K$=SEG(K$,1,LEN(K$)-1)
5150 F$=G$&K$
5160 RETURN

5170 REM
5180 REM          CHECK THE PASSWORD OR FILE NAME IN L$
5190 REM

5200 E0=0
5210 IF LEN(L$)<11 THEN 5240
5220 PRINT "* ***  The name is longer than ten characters. Try again._*"
5230 GO TO 5380
5240 P$=SEG(L$,1,1)
5250 K6=ASC(P$)
5260 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 5290
5270 PRINT "* ***  The name does not start with a letter. Try again._*"
5280 GO TO 5380
5290 FOR I0=1 TO LEN(L$)
5300     P$=SEG(L$,I0,1)
5310     K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
5320     K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
5330     IF K6<=0 THEN 5360
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5340 NEXT IO
5350 GO TO 5390
5360 PRINT "* *** The name contains blanks or non alphanumeric ";
5370 PRINT "characters._*** Try again._*"
5380 EO=1
5390 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 9

```
4000 REM          TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                      OVERLAY     OV5
4050 REM

4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$
4110 H3=1
4120 IF A3<>8 THEN 4140
4130 H3=2
4140 IF A3<>9 THEN 4160
4150 H3=3
4160 GOSUB H3 OF 4180,4310,4490
4170 GO TO 9000

4180 REM     CLOSE ALL FILES AND DELETE LOCAL FILES FROM DISC

4190 CLOSE
4200 PRINT "J***  WARNING  -  LOCAL FILES WILL BE DELETED FROM DISC."
4210 PRINT "***  Have you archived those files you wish saved (Y/N)? ";
4220 INPUT Y$
4230 IF Y$<>"Y" THEN 4300
4240 G$=REP("/LOCAL#",LEN(G$)+1,0)
4250 KILL G$
4260 PRINT @32,26:0
4270 PRINT "J* ***  Execution complete. *JJJ"
4280 PRINT "Press AUTO LOAD key to select other programs."
4290 END
4300 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4310 REM     DIRECTORY OF DATABASE FILES

4320 CALL "UNIT",A9
4330 K$=G$&"/MASTER/DATA#"
4340 PRINT "LDirectory of database files"
4350 CALL "TIME",M$
4360 PRINT "     ";M$
4370 PRINT "J   Master database"
4380 DIRECTORY 0,K$
4390 K$=REP("LOCAL",LEN(G$)+2,6)
4400 PRINT "J   Local databases"
4410 DIRECTORY 0,K$
4420 K$=REP("ARCHIVE",LEN(G$)+2,5)
4430 PRINT "J   Archive databases"
4440 DIRECTORY 0,K$

4450 REM         FORCE FULL PAGE CONDITION

4460 MOVE 0,0
4470 PRINT
4480 RETURN

4490 REM    ROUTINE TO DELETE ARCHIVE FILES

4500 PRINT "JEnter the name of the ARCHIVED file you want to delete ";
4510 PRINT "from the disc. Enter name: __________HHHHHHHHHH";
4520 INPUT L$
4530 IF L$="" THEN 4500
4540 IF L$="!M" THEN 4660
4550 GOSUB 4830
4560 IF E0 THEN 4500
4570 GOSUB 4680
4580 E$=REP("ARCHIVE",LEN(G$)+2,5)
4590 F$=REP("ARCHIVE",LEN(G$)+2,5)
4600 CALL "FILE",A9,E$,K$
4610 IF LEN(K$)>0 THEN 4640
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4620 PRINT "J* *** Archived database was not found on the disc. *"
4630 GO TO 4500
4640 KILL E$
4650 KILL F$
4660 CLOSE
4670 RETURN

4680 REM    SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
4690 REM   L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

4700 K$="/LOCAL/INIT/"
4710 K$=REP(L$,LEN(K$)+1,0)
4720 IF L$<>"" THEN 4750
4730 K$=REP("MASTER",2,5)
4740 K$=SEG(K$,1,LEN(K$)-1)
4750 E$=G$&K$
4760 K$="/LOCAL/DATA/"
4770 K$=REP(L$,LEN(K$)+1,0)
4780 IF L$<>"" THEN 4810
4790 K$=REP("MASTER",2,5)
4800 K$=SEG(K$,1,LEN(K$)-1)
4810 F$=G$&K$
4820 RETURN

4830 REM
4840 REM         CHECK THE PASSWORD OR FILE NAME IN L$
4850 REM

4860 E0=0
4870 IF LEN(L$)<11 THEN 4900
4880 PRINT "* ***  The name is longer than ten characters. Try again. *"
4890 GO TO 5040
4900 P$=SEG(L$,1,1)
4910 K6=ASC(P$)
4920 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 4950
4930 PRINT "* *** The name does not start with a letter. Try again. *"
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4940 GO TO 5040
4950 FOR I0=1 TO LEN(L$)
4960     P$=SEG(L$,I0,1)
4970     K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
4980     K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
4990     IF K6<=0 THEN 5020
5000 NEXT I0
5010 GO TO 5050
5020 PRINT "* *** The name contains blanks or non alphanumeric ";
5030 PRINT "characters._*** Try again._*"
5040 E0=1
5050 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 10

```
4000 REM          TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                    OVERLAY     OV6
4050 REM

4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$
4110 A$="                                        "
4120 A$=A$&A$
4130 B$="HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH"
4140 B$=B$&B$

4150 REM      LIST A LOCAL OF MASTER FILE WITH OR WITHOUT TOTALS

4160 H9=0
4170 IF A3=11 THEN 4190
4180 H9=1
4190 PRINT "JEnter the name of the LOCAL database you want listed, or"
4200 PRINT "enter only a CR if you want the MASTER database listed."
4210 PRINT "Database name? _____HHHHHHHHHH";
4220 INPUT L$
4230 IF L$="!M" THEN 5530
4240 IF L$="" THEN 4270
4250 GOSUB 6350
4260 IF E0 THEN 4190
4270 GOSUB 5600
4280 IF E0 THEN 5580
4290 READ #2,1:I$
4300 IF I$<>"DATABASE IS EMPTY" THEN 4340
4310 PRINT "J* _***  The database is EMPTY. List command ";
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4320 PRINT "aborted._*"
4330 GO TO 5580
4340 CALL "REWIND",2
4350 DIM U$(T4)
4360 IF H9 THEN 4860
4370 PRINT "LJChoose a numeric field to be totaled.J"
4380 H7=1
4390 FOR I1=1 TO T1
4400     IF C(I1,2)=2 THEN 4450
4410     DELETE I$
4420     DIM I$(C(I1,1))
4430     I$=SEG(T$,H7,C(I1,1))
4440     PRINT USING "4X,3D,"")"",2X,62A":I1,I$
4450     H7=H7+C(I1,1)+T8
4460 NEXT I1
4470 PRINT "JField number? ";
4480 GOSUB 6190
4490 IF K7=1 THEN 4470
4500 H8=K6
4510 IF H8=0 THEN 5540
4520 IF H8>0 AND H8<=T1 THEN 4560
4530 PRINT "J*_***  Field number given does not occur in list above.";
4540 PRINT "_try again_*"
4550 GO TO 4370
4560 IF C(H8,2)=2 THEN 4530
4570 PRINT "LChoose a field below to control subtotals.J"
4580 H7=1
4590 FOR I1=1 TO T1
4600     DELETE I$
4610     DIM I$(C(I1,1))
4620     I$=SEG(T$,H7,C(I1,1))
4630     PRINT USING "4X,3D,"")"",2X,62A":I1,I$
4640     H7=H7+C(I1,1)+T8
4650 NEXT I1
4660 PRINT "JField number? ";
4670 GOSUB 6190
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4680 IF K7=1 THEN 4660
4690 H7=K6
4700 IF H7=0 THEN 5540
4710 IF H7>0 AND H7<=T1 THEN 4750
4720 PRINT "J* ***  Field number given does not occur in list above.";
4730 PRINT " Try again. *"
4740 GO TO 4570

4750 REM  J1 IS BEGIN. OF NUM. FIELD, J2 IS BEGIN. OF CONTROL FIELD

4760 I7=1
4770 FOR I1=1 TO T1
4780     IF NOT(H8=I1 OR H7=I1) THEN 4840
4790     IF H8=I1 THEN 4820
4800     J2=I7
4810     GO TO 4840
4820     J1=I7
4830     IF H7=H8 THEN 4800
4840     I7=I7+C(I1,1)+T8
4850 NEXT I1
4860 H4=T4-12 MIN 70
4870 DELETE O$,I$,N$
4880 DIM O$(H4),I$(T4),N$(T4)
4890 C$=SEG(A$,1,H4)
4900 D$=SEG(B$,1,H4)
4910 PRINT "JEnter a ";H4;" character title to be printed with ";
4920 PRINT "listing.  :";C$;D$;
4930 INPUT O$
4940 IF LEN(O$)>H4 THEN 4910
4950 I9=1
4960 J9=0
4970 J8=0
4980 H6=INT((T4-8-LEN(O$))/2)
4990 I$=SEG(X$,1,H6)
5000 H6=T4-H6-LEN(O$)-8
5010 J$=SEG(X$,1,H6)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5020 N$=I$&O$
5030 N$=REP(J$,LEN(N$)+1,0)
5040 N$=REP("PAGE ",LEN(N$)+1,0)
5050 CALL "TIME",M$
5060 M$=SEG(M$,1,15)
5070 H6=INT((T4-23)/2)
5080 I$=SEG(X$,1,H6)
5090 M$=REP(I$,1,0)
5100 H0=100
5110 H1=0
5120 DELETE I$,R$
5130 DIM I$(133),R$(133)
5140 GOSUB 6540
5150 FOR I1=1 TO T2
5160     H0=H0+1
5170     READ #2,I1:U$
5180     IF H9 THEN 5330
5190     DELETE J$,K$
5200     DIM J$(C(H7,1)),K$(C(H8,1))
5210     J$=SEG(U$,J2,C(H7,1))
5220     K$=SEG(U$,J1,C(H8,1))
5230     I4=VAL(K$)
5240     IF I9=0 THEN 5280
5250     R$=J$

5260     REM      R$ IS THE PREVIOUS CONTROL FIELD VALUE
5270     REM       CHECK TO SEE IF ONE SHOULD PRINT SUBTOTALS

5280     GOSUB (R$<>J$)+0 OF 5930

5290     REM              PRINT PAGE HEADINGS WITH TITLES

5300     J8=J8+I4
5310     J9=J9+I4

5320     REM      CHECK TO SEE IF NEW PAGE IS REQUIRED
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5330    GOSUB (H0>A2)+0 OF 6080
5340    PRINT @A1:U$
5350 NEXT I1
5360 IF H9 THEN 5500

5370 REM              PRINT OUT GRAND TOTALS

5380 GOSUB 5930
5390 GOSUB (H0>A2)+0 OF 6080
5400 DELETE I$,L$,K$
5410 K$=STR(J9)
5420 H5=J1+C(H8,1)-LEN(K$)-16
5430 DIM I$(H5+1)
5440 L$="  Grand Total  "
5450 IF H5<0 THEN 5490
5460 I$=SEG(X$,1,H5)
5470 PRINT @A1:I$;L$;K$
5480 GO TO 5500
5490 PRINT @A1:K$;L$
5500 IF A1<>32 THEN 5530
5510 MOVE 0,0
5520 PRINT

5530 REM         EXIT OVERLAY

5540 CLOSE
5550 GOSUB 6610
5560 DELETE A$,B$,C$,D$
5570 GO TO 9000
5580 PRINT USING "4(110(""H"")),"" KK""":
5590 GO TO 5530

5600 REM   ***   OPEN DATABASE FILES

5610 CLOSE
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5620 E0=0
5630 GOSUB 5780
5640 CALL "FILE",A9,E$,K$
5650 IF LEN(K$)>0 THEN 5700
5660 E0=1
5670 PRINT "J* ***  File was not found on disc._*"
5680 PRINT USING "4(110(""H""))","" KK""":
5690 GO TO 5770
5700 OPEN E$;1,"F",K$
5710 OPEN F$;2,"F",K$
5720 READ #1:T1,T2,T9,T8,T4,S1
5730 DELETE C,T$
5740 DIM C(T1,2),T$(T4)
5750 READ #1:C,T$
5760 S$=SEG(X$,1,T8)
5770 RETURN

5780 REM    SUBROUTINE TO CONSTRUCT FILE IDENTIFIERS (F.I.).
5790 REM   L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

5800 K$="/LOCAL/INIT/"
5810 K$=REP(L$,LEN(K$)+1,0)
5820 IF L$<>"" THEN 5850
5830 K$=REP("MASTER",2,5)
5840 K$=SEG(K$,1,LEN(K$)-1)
5850 E$=G$&K$
5860 K$="/LOCAL/DATA/"
5870 K$=REP(L$,LEN(K$)+1,0)
5880 IF L$<>"" THEN 5910
5890 K$=REP("MASTER",2,5)
5900 K$=SEG(K$,1,LEN(K$)-1)
5910 F$=G$&K$
5920 RETURN

5930 REM    PRINT SUBTOTALS
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5940 DELETE I$,L$,K$
5950 K$=STR(J8)
5960 H5=J1+C(H8,1)-LEN(K$)-13
5970 L$="  Subtotal  "
5980 IF H5<0 THEN 6030
5990 DIM I$(H5)
6000 I$=SEG(X$,1,H5)
6010 PRINT @A1:I$;L$;K$;"JJ"
6020 GO TO 6040
6030 PRINT @A1:K$;L$;"JJ"
6040 H0=H0+3
6050 J8=0
6060 R$=J$
6070 RETURN

6080 REM         PRINT PAGE HEADINGS WITH TITLES

6090 IF A1<>32 OR I9 THEN 6120
6100 MOVE 0,0
6110 PRINT " "
6120 H1=H1+1
6130 PRINT @A1:"L";N$;H1
6140 PRINT @A1:M$;"J"
6150 PRINT @A1:T$
6160 H0=5
6170 I9=0
6180 RETURN

6190 REM    SUBROUTINE TO CHECK FOR NON-NUMERIC CHARACTERS

6200 INPUT L$
6210 IF L$="" THEN 6280
6220 K7=0
6230 FOR K5=1 TO LEN(L$)
6240     J$=SEG(L$,K5,1)
6250     IF J$=>"0" AND J$<="9" OR J$=" " THEN 6300
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
6260      PRINT "J* ***  Previous data entered contained non-numeric data."
6270      PRINT "***  Try again._*"
6280      K7=1
6290      GO TO 6320
6300 NEXT K5
6310 K6=VAL(L$)
6320 RETURN

6330 REM         CHECK THE PASSWORD OR FILE NAME IN L$

6340 E0=0
6350 IF LEN(L$)<11 THEN 6380
6360 PRINT "* ***  The name is longer than ten characters. Try again._*"
6370 GO TO 6520
6380 P$=SEG(L$,1,1)
6390 K6=ASC(P$)
6400 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 6430
6410 PRINT "* ***  The name does not start with a letter. Try again._*"
6420 GO TO 6520
6430 FOR I0=1 TO LEN(L$)
6440      P$=SEG(L$,I0,1)
6450      K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
6460      K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
6470      IF K6<=0 THEN 6500
6480 NEXT I0
6490 GO TO 6530
6500 PRINT "* ***  The name contains blanks or non alphanumeric ";
6510 PRINT "characters._***  Try again._*"
6520 E0=1
6530 RETURN

6540 REM     SELECT CHARACTER SIZE

6550 IF RND(0)<0.8 THEN 6600
6560 IF A1<>32 THEN 6600
6570 K9=1+(T4-73<0)+(T4-80<0)+(T4-120<0)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
6580 A2=(K9=1)*56+(K9=2)*54+(K9=3)*34+(K9=4)*30
6590 PRINT @A1,17:K9
6600 RETURN

6610 REM     RESET CHARACTER SIZE

6620 IF RND(0)<0.8 THEN 6660
6630 IF A1<>32 THEN 6660
6640 PRINT @A1,17:4
6650 A2=30
6660 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4280 K$=REP(L$,LEN(K$)+1,0)
4290 IF L$<>"" THEN 4320
4300 K$=REP("MASTER",2,5)
4310 K$=SEG(K$,1,LEN(K$)-1)
4320 E$=G$&K$
4330 K$="/LOCAL/DATA/"
4340 K$=REP(L$,LEN(K$)+1,0)
4350 IF L$<>"" THEN 4380
4360 K$=REP("MASTER",2,5)
4370 K$=SEG(K$,1,LEN(K$)-1)
4380 F$=G$&K$

4390 REM

4400 CALL "FILE",A9,E$,K$
4410 IF LEN(K$)>0 THEN 4450
4420 K1=1
4430 PRINT "J* ***  File was not found on disc._*"
4440 GO TO 4150
4450 OPEN E$;1,"F",K$
4460 OPEN F$;2,"F",K$
4470 READ #1:T1,T2,T9,T8,T4,S1
4480 DELETE C,T$
4490 DIM C(T1,2),T$(T4)
4500 READ #1:C,T$
4510 S$=SEG(X$,1,T8)
4520 DELETE I0
4530 DIM M$(T4),U$(T4),V$(T4),W$(T4),I9(4),K4(10,2),K2(4,3),I0(T1)
4540 RESTORE 4550
4550 DATA 7,9,8,11
4560 READ I9
4570 K2=-200
4580 PRINT "LYou may choose up to four of the fields below to act as"
4590 PRINT " keys which will control the resulting order of items in"
4600 PRINT " the sorted database. The selection of keys is terminated ;
4610 PRINT " by responding to a prompt with a CR."
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 11

```
4000 REM           TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                    OVERLAY   OV7
4050 REM

4060 DELETE H,H0,H1,H2,H3,H4,H5,H6,H7,H8,H9
4070 DELETE I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
4080 DELETE J,J0,J1,J2,J3,J4,J5,J6,J7,J8,J9
4090 DELETE K,K0,K1,K2,K3,K4,K5,K6,K7,K8,K9
4100 DELETE I$,J$,K$,L$,M$,N$,O$,P$,Q$,R$,U$,V$,W$

4110 REM
4120 REM      SORTING OVERLAY
4130 REM

4140 H8=1
4150 PRINT "JEnter the name of the LOCAL database you wish to sort."
4160 PRINT "Database name? _____HHHHHHHHHH";
4170 INPUT L$
4180 IF L$="" THEN 4150
4190 IF L$="!M" THEN 9000
4200 GOSUB 5170
4210 IF E0 THEN 4150

4220 REM   ***   OPEN DATABASE FILES

4230 CLOSE
4240 K1=0

4250 REM                    CONSTRUCT FILE IDENTIFIERS (F.I.).
4260 REM   L$ CONTAINS DATABASE NAME. E$ AND F$ CONTAIN F.I.

4270 K$="/LOCAL/INIT/"
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4940      GO TO 4760
4950      PRINT "Is the field to be sorted in ascending (A) or descend ng"
4960      PRINT " (D) order? ";
4970      INPUT I$
4980      IF I$="" THEN 5000
4990      GO TO POS("AD",I$,1) OF 5020,5040
5000      PRINT "J* ***  Program will only accept an ""A"" or ""D"".  *"
5010      GO TO 4950
5020      K2(I5,3)=1
5030      GO TO 5050
5040      K2(I5,3)=2
5050      K2(I5,1)=I0(K3)
5060      K2(I5,2)=C(K3,1)
5070      IF C(K3,2)=2 THEN 5090
5080      K2(I5,3)=-K2(I5,3)
5090      I8=I8+10
5100      GO TO 5120
5110      I5=4
5120 NEXT I5

5130 REM        CALL OVERLAY OV7A

5140 CALL "UNIT",A8
5150 H$="@DBMPROG/RECORDKEEP/OV7A"
5160 GO TO 540

5170 REM
5180 REM          CHECK THE PASSWORD OR FILE NAME IN L$
5190 REM

5200 E0=0
5210 IF LEN(L$)<11 THEN 5240
5220 PRINT "* ***  The name is longer than ten characters. Try again.  *"
5230 GO TO 5380
5240 P$=SEG(L$,1,1)
5250 K6=ASC(P$)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4620 I7=1
4630 FOR I5=1 TO T1
4640     DELETE I$
4650     DIM I$(C(I5,1))
4660     I$=SEG(T$,I7,C(I5,1))
4670     I0(I5)=I7
4680     PRINT USING "4X,3D,"")"",2X,62A":I5,I$
4690     I7=I7+C(I5,1)+T8
4700 NEXT I5
4710 DELETE I$
4720 L$="primary   secondary tertiary  quanternary"
4730 I8=1
4740 FOR I5=1 TO 4
4750     N$=SEG(L$,I8,I9(I5))
4760     PRINT "JEnter the field number to be used as the ";N$;" key: ";
4770     INPUT I$
4780     IF I$="" THEN 5110

4790     REM        CHECK FOR NON-NUMERIC CHARACTERS

4800     FOR I7=1 TO LEN(I$)
4810         J$=SEG(I$,I7,1)
4820         IF J$=>"0" AND J$<="9" OR J$=" " THEN 4860
4830         PRINT "J* ***  Previous data entered contained non-numer c data
4840         PRINT "***  Try again._*"
4850         GO TO 4760
4860     NEXT I7

4870     REM

4880     K3=VAL(I$)
4890     IF K3>0 THEN 4920
4900     CLOSE
4910     GO TO 9000
4920     IF K3<=T1 THEN 4950
4930     PRINT "J*_***  The field number is not between 1 and ";T1;"._*"
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5260 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 5290
5270 PRINT "* ***  The name does not start with a letter. Try again._*"
5280 GO TO 5380
5290 FOR I0=1 TO LEN(L$)
5300     P$=SEG(L$,I0,1)
5310     K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
5320     K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
5330     IF K6<=0 THEN 5360
5340 NEXT I0
5350 GO TO 5390
5360 PRINT "* ***  The name contains blanks or non alphanumeric ";
5370 PRINT "characters._***  Try again._*"
5380 E0=1
5390 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 12

```
4000 REM            TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                        OVERLAY    OV7A
4050 REM

4060 IF K2(1,1)<-100 THEN 5130

4070 REM      BEGINNING OF SORTING ALGORYTHM
4080 REM

4090 PRINT "J* Sorting has begun. *"

4100 REM
4110 REM    *** Q1    INITIALIZE

4120 H1=1
4130 H2=T2
4140 S=-100
4150 K5=0
4160 IF T2<=H8 THEN 4920

4170 REM    *** Q2    BEGIN NEW STAGE

4180 H4=H2+1
4190 H3=H1
4200 READ #2,H1:U$

4210 REM    *** Q3    COMPARE K AND Ki

4220 IF H3=H2 THEN 4300
4230 H3=H3+1
4240 IF H1=H2 THEN 4810
4250 IF H4=H1 AND H3=H1+1 THEN 4430
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4260 READ #2,H3:V$
4270 M$=V$
4280 GOSUB 5170
4290 IF K6 THEN 4210

4300 REM    *** Q4    COMPARE K AND Kj

4310 IF H4=H1 THEN 4510
4320 H4=H4-1
4330 READ #2,H4:W$
4340 M$=W$
4350 GOSUB 5170
4360 IF NOT(K6) THEN 4300

4370 REM    *** Q5    TEST i AND j

4380 IF NOT(H4>H3) THEN 4480

4390 REM    *** Q6    EXCHANGE RECORDS

4400 WRITE #2,H3:W$
4410 WRITE #2,H4:V$
4420 GO TO 4210

4430 REM

4440 WRITE #2,H1:V$
4450 WRITE #2,H3:U$
4460 H1=H1+1
4470 GO TO 4170

4480 REM

4490 WRITE #2,H1:W$
4500 WRITE #2,H4:U$
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4510 REM    *** Q7    PUSH ON STACK

4520 IF H2-H4=>H4-H1 THEN 4650
4530 IF H8=>H4-H1 AND H4-H1=>H2-H4 THEN 4810
4540 IF H4-H1>H8 AND H8=>H2-H4 THEN 4750
4550 K5=K5+1
4560 IF K5=1 THEN 4610
4570 FOR I1=K5-1 TO 1 STEP -1
4580      K4(I1+1,1)=K4(I1,1)
4590      K4(I1+1,2)=K4(I1,2)
4600 NEXT I1
4610 K4(1,1)=H1
4620 K4(1,2)=H4-1
4630 H1=H4+1
4640 GO TO 4170
4650 IF H8=>H2-H4 AND H2-H4=>H4-H1 THEN 4810
4660 IF H2-H4>H8 AND H8=>H4-H1 THEN 4770
4670 K5=K5+1
4680 IF K5=1 THEN 4730
4690 FOR I1=K5-1 TO 1 STEP -1
4700      K4(I1+1,1)=K4(I1,1)
4710      K4(I1+1,2)=K4(I1,2)
4720 NEXT I1
4730 K4(1,1)=H4+1
4740 K4(1,2)=H2
4750 H2=H4-1
4760 GO TO 4170
4770 H1=H4+1
4780 GO TO 4170
4790 H2=H4-1
4800 GO TO 4170

4810 REM    *** Q8    PULL OFF STACK

4820 IF K5<=0 THEN 4920
4830 H1=K4(1,1)
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4840 H2=K4(1,2)
4850 K5=K5-1
4860 IF K5<=0 THEN 4170
4870 FOR I1=1 TO K5
4880      K4(I1,1)=K4(I1+1,1)
4890      K4(I1,2)=K4(I1+1,2)
4900 NEXT I1
4910 GO TO 4170

4920 REM    *** Q9    INSERTION SORT

4930 IF H8<=1 THEN 5130
4940 IF T2<2 THEN 5130
4950 H4=1
4960 H4=H4+1
4970 IF H4>T2 THEN 5130
4980 H3=H4-1
4990 READ #2,H4:U$
5000 READ #2,H3:M$
5010 GOSUB 5170
5020 IF K6 THEN 4960
5030 WRITE #2,H3+1:M$
5040 H3=H3-1
5050 IF H3>0 THEN 5080
5060 WRITE #2,1:U$
5070 GO TO 4960
5080 READ #2,H3:M$
5090 GOSUB 5170
5100 IF NOT(K6) THEN 5030
5110 WRITE #2,H3+1:U$
5120 GO TO 4960

5130 REM    *** Q10    END OF SORTING ALGORYTHM
5140 REM

5150 CLOSE
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
5160 GO TO 9000

5170 REM       SUBROUTINE TO COMPARE KEYS
5180 REM  K2(i,3)  *  i= 1 THEN ALPHA,ASCEND  * i= 2 THEN ALPHA,DESCEND
5190 REM           *  i=-1 THEN NUM,ASCEND    * i=-2 THEN NUM,DESCEND

5200 K6=0
5210 FOR I2=1 TO 4
5220     IF K2(I2,1)<-100 THEN 5420
5230     DELETE N$,K$
5240     DIM N$(K2(I2,2)),K$(K2(I2,2))
5250     N$=SEG(M$,K2(I2,1),K2(I2,2))
5260     K$=SEG(U$,K2(I2,1),K2(I2,2))
5270     IF K2(I2,3)<0 THEN 5340

5280     REM        ALPHANUMERIC FIELD COMPARISON

5290     IF K$=N$ THEN 5430
5300     K6=K$>N$
5310     IF K2(I2,3)=1 THEN 5440
5320     K6=K$<N$
5330     GO TO 5440

5340     REM        NUMERIC FIELD COMPARISON

5350     H5=VAL(K$)
5360     H6=VAL(N$)
5370     IF H5=H6 THEN 5430
5380     K6=H5>H6
5390     IF K2(I2,3)=-1 THEN 5440
5400     K6=H5<H6
5410     GO TO 5440
5420     I2=4
5430 NEXT I2
5440 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 13

```
4000 REM            TARGET LINE FOR OVERLAY APPEND
4010 REM
4020 REM ------------------------------------------------------------
4030 REM
4040 REM                        OVERLAY     OV0
4050 REM

4060 DELETE A,B,A$,B$,C$,D$,L$,K$,P$
4080 PRINT "LIRECORDKEEPING & SORTED REPORT GENERATOR"
4090 PRINT "JThis program will maintain an inventory record and ";
4100 PRINT "generate sorted reports at the user's request. At initial ";
4110 PRINT "usage, the user must initialize the disc file through use ";
4120 PRINT "of the file initialization "
4130 PRINT "routine."
4140 CALL "UNIT",A9
4150 PRINT "JEnter the desired database family: _____";
4160 PRINT "HHHHHHHHHHHHHHHHHHHHH";
4170 INPUT K$
4180 IF K$="" THEN 4150
4190 K1=POS(K$,":",1)
4200 IF K1<>1 THEN 4240
4210 PRINT "* ***  A family name must preceed any password.";
4220 PRINT " Try again._*"
4230 GO TO 4150
4240 IF K1<LEN(K$) THEN 4280
4250 PRINT "* ***  A password must follow the colon, if present.";
4260 PRINT " Try again._*"
4270 GO TO 4150
4280 IF K1>0 THEN 4310
4290 L$=K$
4300 GO TO 4320
4310 L$=SEG(K$,1,K1-1)
4320 J$="Family name"
4330 GOSUB 4720
4340 IF E0 THEN 4150
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4350 IF K1<=0 THEN 4400
4360 L$=SEG(K$,K1+1,LEN(K$)-K1)
4370 J$="Password"
4380 GOSUB 4720
4390 IF E0 THEN 4150
4400 G$="@DATABASE/"&K$
4410 F$=G$&"/MASTER/INIT"
4420 CALL "FILE",A9,F$,K$
4430 IF LEN(K$)>0 THEN 4460
4440 PRINT "J* ***  No files found under that family name._*"
4450 GO TO 4150
4460 DELETE X$
4470 OPEN F$;1,"R",K$
4480 CLOSE
4490 DIM Y$(1),X$(135),W$(22),S$(70),R$(55),P$(55),Q$(75)
4500 X$="                                                                    "
4510 O$="                                                                    "
4520 X$=REP(O$,1,0)
4530 X$=REP("            ",1,0)

4540 REM   ***   OUTPUT DEVICE CHOICE
4550 REM   ***   PRIMARY ADDRESS 32 FOR SCREEN AND 55 FOR PRINTER

4560 A1=32
4570 A2=30
4580 H1=0
4590 PRI "JDo you want reports printed to screen(0) or printer (1):  ";
4600 INPUT H1
4610 IF H1<>1 AND H1<>0 THEN 4590
4620 IF H1=0 THEN 4660
4630 PRINT "JEnter the primary address of the printer: ";
4640 INPUT A1
4650 A2=60
4660 DELETE U$,Z$
4670 DIM U$(133),Z$(75),A(12)
4680 H$=""
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
4690 DATA 1,1,2,6,3,4,4,5,5,7,6,5
4700 READ A
4710 GO TO 9000

4720 REM
4730 REM          CHECK THE PASSWORD OR FILE NAME IN L$
4740 REM

4750 E0=0
4760 IF LEN(L$)<11 THEN 4790
4770 PRINT "* ***  ";J$;" is longer than ten characters. Try again._*
4780 GO TO 4930
4790 P$=SEG(L$,1,1)
4800 K6=ASC(P$)
4810 IF K6>64 AND K6<91 OR K6>96 AND K6<123 THEN 4840
4820 PRINT "* ***  ";J$;" does not start with a letter. Try again._*"
4830 GO TO 4930
4840 FOR I0=1 TO LEN(L$)
4850     P$=SEG(L$,I0,1)
4860     K6=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",P$,1)
4870     K6=K6+POS("abcdefghijklmnopqrstuvwxyz",P$,1)
4880     IF K6<=0 THEN 4910
4890 NEXT I0
4900 GO TO 4940
4910 PRINT "* ***  ";J$;" contains blanks or non alphanumeric ";
4920 PRINT "characters._*** Try again._*"
4930 E0=1
4940 RETURN
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

FILE 14

```
100 REM   ***   BEGINING OF THE 4907 DISC FILE INITIALIZATION

110 CALL "MOUNT",0,""

120 REM   ***   REQUEST LIBRARY TITLE FOR DATABASE DEFINITION

130 DELETE A$
140 DIM A$(10)
150 PRINT "LI        TAPE DATABASE TO DISC DATABASE"
160 PRINT "I              CONVERSION PROGRAMJJ"
170 PRINT "JInput database family name: ";
180 INPUT A$
190 IF A$="" THEN 170
200 B$="@DATABASE/"&A$
210 B$=REP("/MASTER",LEN(B$)+1,0)

220 REM   ***   DETERMINE IF THE NAME GIVEN IS PREVIOUSLY DEFINED

230 C$=B$&"/INIT"
240 CALL "FILE",0,C$,Z$
250 IF LEN(Z$)=0 THEN 280
260 PRINT "J* ***  Files already exist under that family name._*"
270 GO TO 170

280 REM   ***   DEFINE FILEDEF TO CONTAIN RECORD DEFINITION
290 REM   ***   DEFINE DATABASE TO CONTAIN RECORDS

300 DIM K$(1)
310 L$="BU"
320 PRINT "JDo you want this data base to be a private file? (Y/N): "
330 INPUT K$
340 GO TO POS("DYN",K$,1) OF 320,360,440
350 GO TO 320
360 PRINT "JEnter a file Password (1 to 10 characters must start with
370 PRINT "  a leter but the remaining characters can be alphanumeric. "
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
380 PRINT "Password is: ";
390 INPUT J$
400 IF J$="" THEN 360
410 J$=REP(":",1,0)
420 B$=REP(J$,POS(B$,"/MASTER",9),0)
430 L$="BR"
440 FIND 1
450 F$="           "
460 INPUT @33:K1
470 I1=K1+1
480 DIM C(I1,2),D(K1,2)
490 INPUT @33:D,K2,T9,T8,K4,S1,U$,S$
500 C(1,1)=5
510 C(1,2)=1
520 FOR K0=1 TO K1
530     C(K0+1,1)=D(K0,1)
540     C(K0+1,2)=D(K0,2)
550 NEXT K0
560 T4=72
570 DIM T$(72)
580 T$=SEG(U$,1,72)
590 T2=0

600 REM   CREATE DISC FILE AND TRANSFER TAPE DATA TO DISC FILE

610 C$=B$&"/DATA"
620 CREATE C$,L$;1,T4+5
630 OPEN C$;1,"F",K$
640 WRITE #1,1:"DATABASE IS EMPTY"
650 CALL "REWIND",1

660 REM

670 DELETE V$
680 DIM V$(T4)
690 PRINT "JEnter the number of data files on tape: ";
```

TITLE	ABSTRACT NO:
RECORDKEEP II	51/07-6108/1

```
700 INPUT K9
710 FOR K0=2 TO K9
720     ON EOF (0) THEN 830
730     FIND K0
740     INPUT @33:V$
750     IF LEN(V$)=0 THEN 740
760     T2=T2+1
770     E$=STR(T2)
780     G$=SEG(F$,1,5-LEN(E$))
790     E$=REP(G$,1,0)
800     V$=REP(E$,1,5)
810     WRITE #1,T2:V$
820     GO TO 740
830 NEXT K0

840 REM     WRITE HEADER FILE ("INIT") FOR DISC DATABASE ("DATA")

850 L0=2*T1*10+T4+6 MAX 60
860 C$=B$&"/INIT"
870 CREATE C$,"U";L0,0
880 OPEN C$;2,"F",K$
890 WRITE #2:T1,T2,T9,T8,T4,S1
900 WRITE #2:C,T$
910 CLOSE
920 PRINT "J* *** CONVERSION COMPLETE._*"
930 END
```

TEKTRONIX TAPE 19 F 25-26

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0
ORIGINAL DATE: 1/80 / REVISION DATE:	MEMORY REQUIREMENT: 8K
AUTHOR: Robert Cope & Michael Bowman, Atlantic Analysis Corporation, Norfolk, VA	PERIPHERALS: Hewlett Packard 9835 or 9845

ABSTRACT

Statements: 70 & 77
Files: 1 ASCII Program
1 ASCII Text

This program transfers messages input from the keyboard of an H-P 9835 to a TEKTRONIX 4051 and vice versa. The purpose is to demonstrate the capability of the H-P 9835 and TEKTRONIX 4051 to communicate using the standard IEEE-488 Interface bus and to demonstrate the interrupt capabilities and limitations of both computers. The configuration simulates an interconnection between a TEKTRONIX 4054 and a Hewlett Packard 9845 which were not available at the time the evaluation was conducted.

Communications using the 488 bus will make it possible to offload the graphics processing load in ECLIPS tactical display systems to an intelligent graphics display which will free the other computer for message processing and textual information display. The interrupt capabilities of the machines will make it possible for each processor to devote full time to their respective tasks, being interrupted only to service operation requests or intercomputer communications needs on a time available basis ensuring that the high priority message processing is never neglected.

The complete evaluation is included as part of the documentation.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0

Operation

The attached report contains the equipment configuration, operating instructions and methodology. Please refer to it.

Tape Structure

The 4050 Program and the H-P program are both included on tape. However, to input and display the latter on the 4050 Screen, a 4051R06 Editor ROM Pack is required or the following program will have to be keyed in to read it in and display it.

```
500 INIT
510 DIM B$(8000)
520 ON EOF (0) THEN 600
530 B$=""
540 C$=CHR(13)
550 FIND 2
560 INPUT @33:A$
570 A$=A$&C$
580 B$=B$&A$
590 GO TO 560
600 PRINT B$
610 END
```

TITLE	ABSTRACT NO:
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0

```
3 GO TO 100
4 GOSUB 360

100 REM 9835- 4051 COMS 488

110 INIT
120 SET KEY
130 ON SRQ THEN 470
140 DIM A$(288),B$(288),I$(80),E$(80)
142 F=0
144 T=0
146 R=0
150 C$="DISPLAY FULL - MAKE COPY - PRESS KEY TO CONT"
155 E$=" "
160 GOSUB 510
170 PRINT "↑"
171 FOR I=1 TO 5
180     PRINT "MAIN PROGRAM EXECUTING"
190 NEXT I
200 MOVE 0,76
210 PRINT A$
220 MOVE 0,45
230 PRINT B$
240 IF F<>1 THEN 300
250 MOVE 0,11
260 PRINT C$
262 PRINT "G"
264 FOR X=1 TO 100
266 NEXT X
270 F=0

280 REMUT C

290 GOSUB 510
300 IF T<>1 THEN 320
310 GOSUB 420
```

TITLE	ABSTRACT NO:
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0

```
320 IF R<>1 THEN 335
330 GOSUB 560
335 FOR I=1 TO 500
336 NEXT I
340 PAGE
350 GO TO 170

360 REM SUB KEY INTERUPT

361 MOVE 0,9
370 INPUT K$
380 T=1
390 IF LEN(A$)+LEN(K$)<288 THEN 410
400 F=1
410 GO TO 170

420 REM SUB TRANSMIT

430 PRINT @20:K$
440 A$=A$&K$
450 T=0
460 RETURN

470 REM SRQ INTERUPT

480 POLL M,W;20
490 R=1
500 RETURN

510 REM SUB STRING CLEAR

520 A$="TRANSMITTED TO 9835__"
530 B$="RECEIVED FROM 9835__"&E$
540 E$=" "
550 RETURN
```

TITLE	ABSTRACT NO:
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0

```
560 REM SUB RECEIVE

570 INPUT @20:I$
580 IF LEN(B$)+LEN(I$)<288 THEN 630
590 F=1
600 E$=I$
610 R=0
620 RETURN
630 B$=B$&I$
640 R=0
650 RETURN
660 END
```

TITLE	ABSTRACT NO:
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0

```
10   ! 9835-4051 COMS 488
20   ON INT #7,15 CALL Tekservice
30   CONTROL MASK 7;"00110000"
40   CARD ENABLE 7
50   COM C,E,L,D,K$[80],P,Tekin$[80]
60 SET TIMEOUT 7;100
70   P=0
80   K$=""
90   C=E=5
100  L=2
110  D=11
120  ON KBD CALL Keyservice
130 DISP "MAIN PROGRAM EXECUTING"
140  REM
150  REM
160  REM
170  DISP CHR$(13)
180  GOTO 130
190 END
200  SUB Tekservice
210  BEEP
220  IF TIME OUT(7)>0 THEN 270
230  STATUS 7;Stat
240  IF BIT(Stat,4) THEN CALL Input
250  IF BIT(Stat,5) THEN CALL Output
```

TITLE	ABSTRACT NO:
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0

```
260  GOTO 290
270  DISP "TIMEOUT - MESSAGE NOT SENT"
280  STOP
290  CARD ENABLE 7
300  SUBEND
310 SUB Clear(S)
350  IF S<1 THEN 430
360 FOR I=2 TO 10
370  PRINT USING "#,K";CHR$(27)&"H"
380  PRINT LIN(I);SPA(80)
390  NEXT I
400  L=2
420  SUBEXIT
430  FOR I=11 TO 20
440  PRINT USING "#,K";CHR$(27)&"H"
450  PRINT LIN(I);SPA(80)
460  NEXT I
470  D=11
490  SUBEND
500  SUB Input
510  COM C,E,L,D,K$[80],P,Tekin$[80]
520  ENTER 7;Tekin$
530  PRINT USING "#,K";CHR$(27)&"H"
550  PRINT LIN(L);Tekin$
580  L=L+1
600  IF L>10 THEN CALL Clear(1)
```

TITLE	ABSTRACT NO:
TEKTRONIX/HEWLETT PACKARD 488 INTERFACE	51/00-6117/0

```
610  SUBEND
620  SUB Output
630  COM C,E,L,D,K$[80],P
640  OUTPUT 7;K$
650  P=0
660  DISP CHR$(12)
670  PRINT USING "#,K";CHR$(27)&"H"
671  PRINT SPA(80)
675  PRINT USING "#,K";CHR$(27)&"H"
680  PRINT LIN(D);K$
690  K$=""
700  D=D+1
710  IF D>20 THEN CALL Clear(0)
720  SUBEND
730  SUB Keyservice
740 COM C,E,L,D,K$[80],P
750  P=P+1
760  K$[P,P]=KBD$
770  PRINT USING "#,K";CHR$(27)&"H"
780  PRINT SPA(80)
785  PRINT USING "#,K";CHR$(27)&"H"
790  PRINT K$
791  IF LEN(K$)>79 THEN 810
800  IF NUM(K$[P,P])<>255 THEN SUBEXIT
810  REQUEST 7;64
820  SUBEND
```

Included as part of 4050 Series
Applications Library Documentation
Abstract #51/00-6117/0

ATLANTIC ANALYSIS CORPORATION

AAC-91-79-8

Evaluation of An IEEE-488 Standard Interface For Inter Computer Communications Between The Hewlett-Packard 9835 And The Tektronix 4051 Computers

Prepared for:
Naval Ocean Systems Center
Code 831

Prepared by:
Mr. Robert J. Cope
Mr. Michael E. Bowman

Contract:
Office of Naval Research
N00014-79-C-0347

November 14, 1979

5 KOGER EXECUTIVE CENTER • SUITE 219 • NORFOLK, VIRGINIA 23502
(804) 461-1980

TABLE OF CONTENTS

		Page
1.	PURPOSE	1
2.	EQUIPMENT CONFIGURATION	1
3.	OPERATIONAL DESCRIPTION	1
4.	SOFTWARE CONSIDERATIONS	3
	4.1 IEE-488 Bus	3
	4.2 Tektronix 4051	4
	4.3 Hewlett-Packard 9835	4
5.	PROGRAM DESCRIPTIONS	5
	5.1 Tektronix 4051	5
	5.2 HP-9835	9
6.	CONCLUSIONS	13
APPENDIX		

EVALUATION OF AN IEEE-488 STANDARD INTERFACE FOR INTER COMPUTER COMMUNICATIONS BETWEEN THE HEWLETT-PACKARD 9835 AND THE TEKTRONIX 4051 COMPUTERS

1. PURPOSE

The purpose of this evaluation is to demonstrate the capability of the Hewlett-Packard 9835 and Tektronix 4051 to communicate using the standard IEEE-488 Interface bus and to demonstrate the interrupt capabilities and limitations of both computers. This configuration simulates an interconnection between Tektronix 4054 and Hewlett-Packard 9845 which were not available at the time the evaluation was conducted. Communications using the 488 bus will make it possible to off load the graphics processing load in ECLIPS tactical display systems to an intelligent graphics display such as the Tektronix 4054. This will in turn free the Hewlett Packard 9835 or 9845 computer for message processing and textual information display. The interrupt capabilities of these machines will make it possible for each processor to devote full time to their respective tasks, being interrupted only to service operation requests or intercomputer communications needs on a time available basis ensuring that the high priority message processing is never neglected. The facilities of the NOSC Procal Lab as well as technical support provided by Mr. Jim Grant (NOSC), Hewlett-Packard and Tektronix were instrumental in the successful completion of this project.

2. EQUIPMENT CONFIGURATION

For the purposes of this evaluation, a Hewlett-Packard 9835 desktop computer was connected to a Tektronix 4051 graphic system using the Hewlett-Packard 98034A HP-IB Interface (See Figure 1). The HB-IB bus is based on the IEEE-488 Standard. In addition, each computer was equipped with a printer for hard copy program listing. The HP-9835 was equipped with the 98332A I/O ROM Package. The HP-IB Interface was set to select code 7, the talk/listen address was set to 20, and the system controller switch was in the (off) position.

3. OPERATIONAL DESCRIPTION

The software written for this evaluation was designed to transfer messages input from the keyboard on one computer to the display screen of the other computer. Listings are included in the Appendix. On the 4051 screen, the transmitted messages are displayed on the upper half and received messages are displayed on the lower half of the display. The order of display is reversed on the 9835. Messages sent from the 9835 may be 80 characters in length. As the message is typed in it is displayed at the top of the screen. Pressing the (execute) key causes the message to be sent to the 4051 and also displayed on the 9835 screen.

The lack of a keyboard interrupt capability on the 4051 makes it necessary to press special function key (1) to interrupt the processor for message processing. If the message to be sent is less than 25 characters, the system processing need not be interrupted until the message has been typed in because the keyboard has a 26 character buffer. The message is typed in and special function key (1) is pressed. The

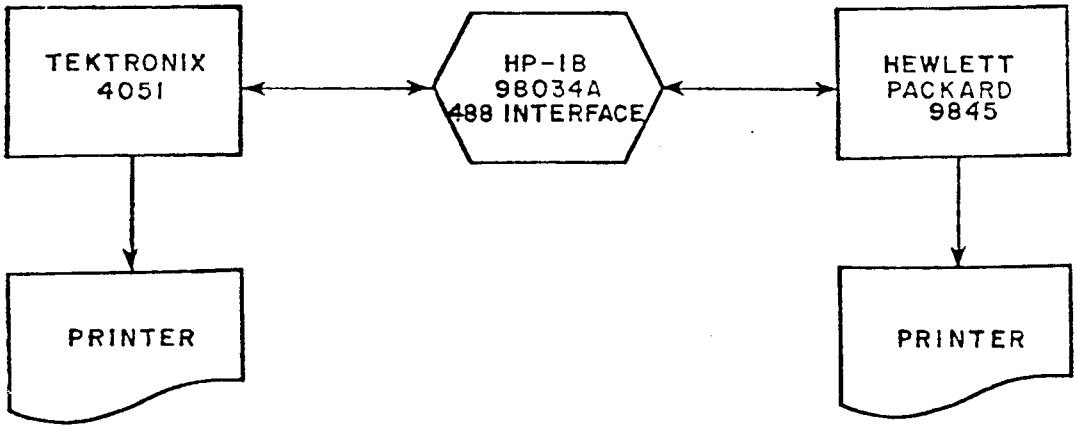

TEKTRONIX
4051
HP-IB
98034A
488 INTERFACE
HEWLETT
PACKARD
9845
PRINTER
PRINTER

Figure 1.

message will appear at the bottom of the screen and may be edited at this time. To send the message the (return) key is pressed. The message will then be transmitted and displayed.

If the message is greater than 26 characters then special function key (1) is pressed first; then the message, up to 80 characters, is typed in. The (return) key is pressed to transmit and display the message.

To demonstrate that a main program is operating until interrupted, the Tektronix 4051 periodically clears the screen and re-displays the message "MAIN PROGRAM EXECUTING" along with the current transmitted and received messages. The HP-9835 flashes the same message on the bottom display line of the CRT. Since the 9835 display screen is a refresh type display it is not necessary to periodically clear the entire screen.

4. SOFTWARE CONSIDERATIONS

In order to implement communications on the IEEE-488 bus it is important to understand the bus architecture as well as the interrupt capabilities and limitations of each computer.

4.1 IEEE-488 Bus

The IEEE-488 bus consists of 16 lines divided into three functional groups; an eight line data bus, three line transfer bus, and a five line management bus. Data and device address commands are transferred bit parallel, byte serial on the data bus. The transfer bus is used by devices on the bus to time the transfer of data. Data is transferred at the rate of the slowest device on the bus. The maximum transfer rate is 256K bytes/second or 2.048 million bits/second. This may be compared to the RS-232 interface with an upper transfer limit of approximately 9600 bits/second. The management bus is used by the system controller to configure the bus and by devices on the bus to gain the attention of the system controller.

For a complete description of the bus the reader is directed to the IEEE Standard #488-1975, the Tektronix 4050 series Reference Manual, and the Hewlett-Packard HP-IB Interface Installation and Service Manual. The latter two references will describe the bus functions implemented on the respective computers.

Tektronix and Hewlett-Packard use negative = true or 1 logic on the 488 bus. This means that a negative voltage is applied to a line in order to pull the line true. The bus is controlled by a "system controller" and all devices on the bus are either talkers or listeners, each having a talk/listen address. The system controller designates a talker or a listener by pulling true the ATN (attention) line on the management bus and then sending the appropriate addresses on the data bus. The designation as a talker or listener will cause the addressed devices to execute either an input or output operation as required. If a device needs the attention of the system controller it will pull the SRQ (service request) line on the management bus true. This causes the system controller to poll the devices on the bus to determine the source of the SRQ. In response to the poll the requesting device sends a status byte which the system controller uses to determine the proper action to take.

4.2 Tektronix 4051

Unlike the HP-9835, the 4051 does not have the ability to relinquish system control of the bus to another controller. Therefore, it is important to keep in mind that the 9835 must be simply a device on the bus. In this test program four program statements are involved in the handling of bus communications and interrupts. These are:

ON SRQ THEN - This statement sets an interrupt condition which calls an interrupt service routine whenever the SRQ line is pulled true.

POLL M, W: 20 - This statement is usually part of the interrupt service routine. On execution of this statement the SRQ will be reset and the list of device addresses following the colon will be polled one at a time. The order in the address list of the requesting device will be returned in variable M and the status byte sent in answer to the poll will be returned in variable W.

INPUT @20: A$ - This statement is executed when the 4051 requests input from device 20. The device will be sent its talk address and the data sent by the device will be returned in A$.

PRINT @20: A$ - This statement is executed when the Tektronix 4051 needs to send data to device 20. The device (HP-9835) will be sent its listen address and the data in A$ will be transferred to the device.

4.3 Hewlett-Packard 9835

The HP-9835 with extended I/O ROM is a versatile user of the 488 interface capable of acting as a system controller or a device. In this test program there are eight program statements involved with handling interrupts and bus communications. These are:

ON INT #7, 15 call service - This statement sets an interrupt condition with a priority of 15 (HIGH) which will call the subroutine named service when an interrupt signal is received on select code 7.

CONTROL MASK 7; "00110000" - This statement indicates to the interface card at select code 7 the conditions under which an interrupt should be generated. In this case bits 4 and 5 are set which indicates that an interrupt should be generated when addressed as a talker or a listener on the 488 bus.

Card Enable 7 - This statement sets the interface at select code 7 to respond to interrupts. It must be executed after each interrupt is generated to enable another interrupt. This statement must not be executed more than once between interrupts or system failure will result.

ON KBD CALL service - This statement sets the keyboard interrupt condition which calls the subroutine named service every time a key is pressed and also saves the value of the key pressed in an 80 character buffer called KBD$.

STATUS 7; S - This statement queries the interface at select code 7 and returns a status byte in variable S which indicates, in this case, whether the computer has been addressed as a talker or a listener.

ENTER 7; T$ - This statement is executed when data is to be received on the 488 bus at select code 7. The data on the bus is returned in T$.

OUTPUT 7; K$ - This statement is executed when data is to be transmitted over the 488 bus with K$ containing the data.

REQUEST 7; 64 - This statement is executed when the 9835 requires the attention of the system controller. The SRQ line will be pulled true and the response byte, 64, will be sent in answer to a serial poll by the system controller. The response byte must have bit six set, that is, have a value in the range 64 to 127 in order for the SRQ to be pulled true. If a request is executed with bit six of the response byte = 0 then the SRQ line will be reset to logical 0.

5. PROGRAM DESCRIPTIONS

The Appendix contains program listings of the software developed during this evaluation. Each program is divided into three main sections. The main program contains the code for start up and the executive loop which simulates an ongoing program in execution. The interrupt service routines contain the code executed only when an interrupt condition exists and processing outside the main program is required. The subroutines contain the code for those routines called for communications handling or display management.

5.1 Tektronix 4051

The main program in the Tektronix 4051 (See Figure 2) sets the interrupt conditions, initializes variables and then enters into the executive loop. The executive loop prints the screen, checks the status of the screen full flag, clears the screen and then begins again. This continues until an interrupt occurs.

When special function key (1) is pressed an unconditional branch is executed to the special function key service routine (See Figure 3). The SFK routine inputs the message from the keyboard, sets the transmit flag, checks for a full screen and continues execution at the beginning of the executive loop. Note that this is not truly a keyboard interrupt since the 4051 does not provide that capability.

When the SRQ line has been set to true by the HP-9835, the Tektronix 4051 generates a true interrupt which branches to the SRQ service routine (See Figure 3). The SRQ service routine polls the bus, sets the receive flag and returns to the program line where the interrupt occurred.

If the screen full flag is set then the screen full message is printed and a short wait is entered to allow hard copy. The display strings are cleared and if a message has been received when the screen was full, that message is added to the display string. The screen full flag is reset to 0 and the transmit flag is checked next (See Figure 4).

If the transmit flag is set, this indicates a message has been input from the keyboard and execution branches to the transmit subroutine (See Figure 4).

The transmit subroutine transmits the message entered from the keyboard to the 488 bus and adds the message to the display string. The message will be displayed on the next pass made through the executive loop. The transmit flag is reset to 0 and execution returns to the executive loop where the receive flag is checked.

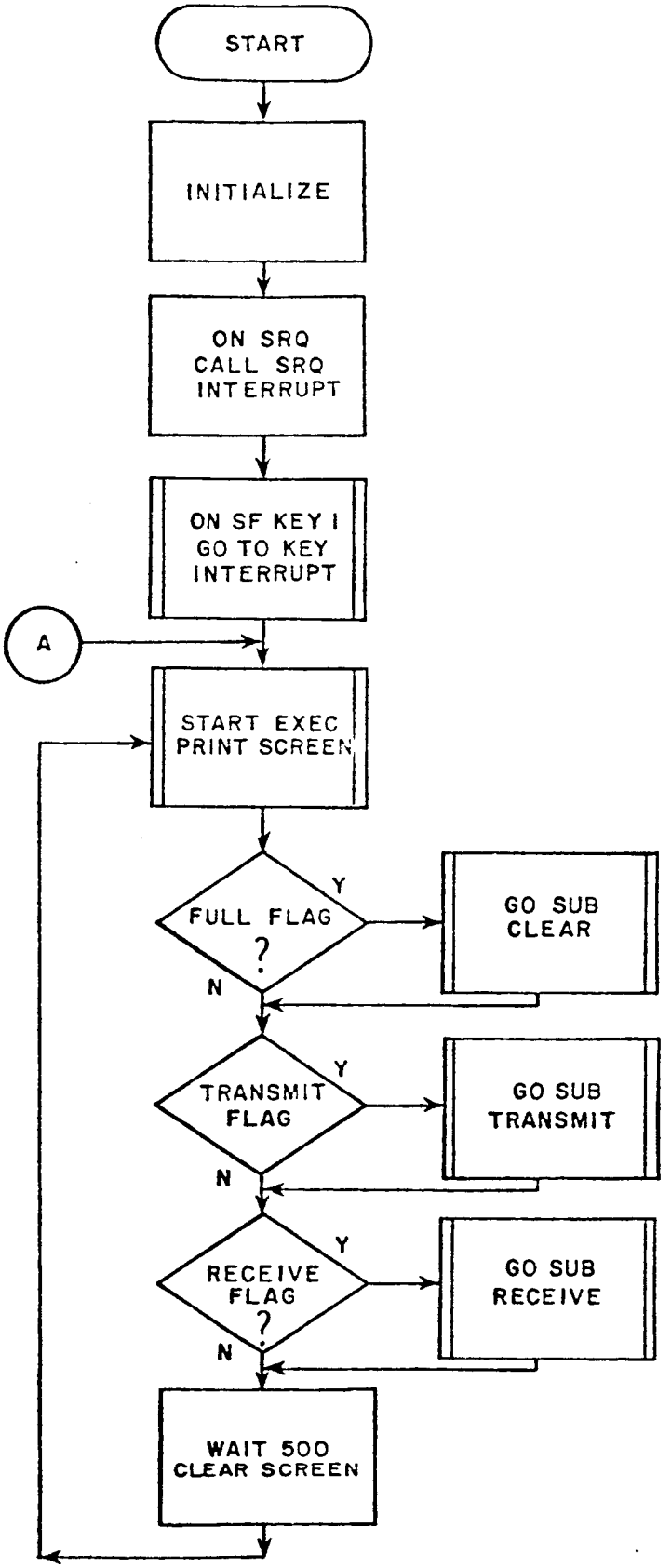


Figure 2. TEK 4051 MAIN PROGRAM

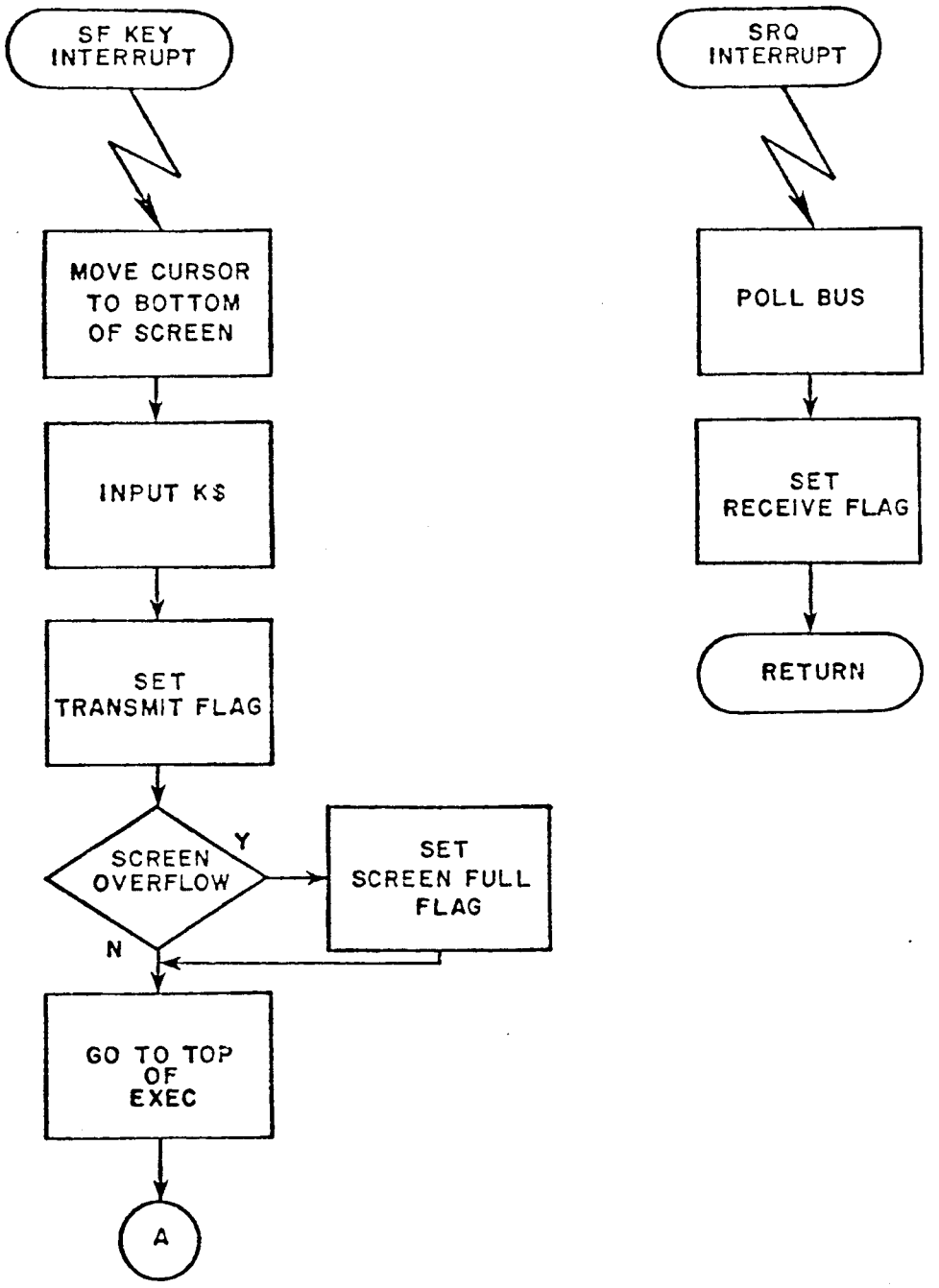


Figure 3. TEK 4051 INTERRUPT SERVICE ROUTINES

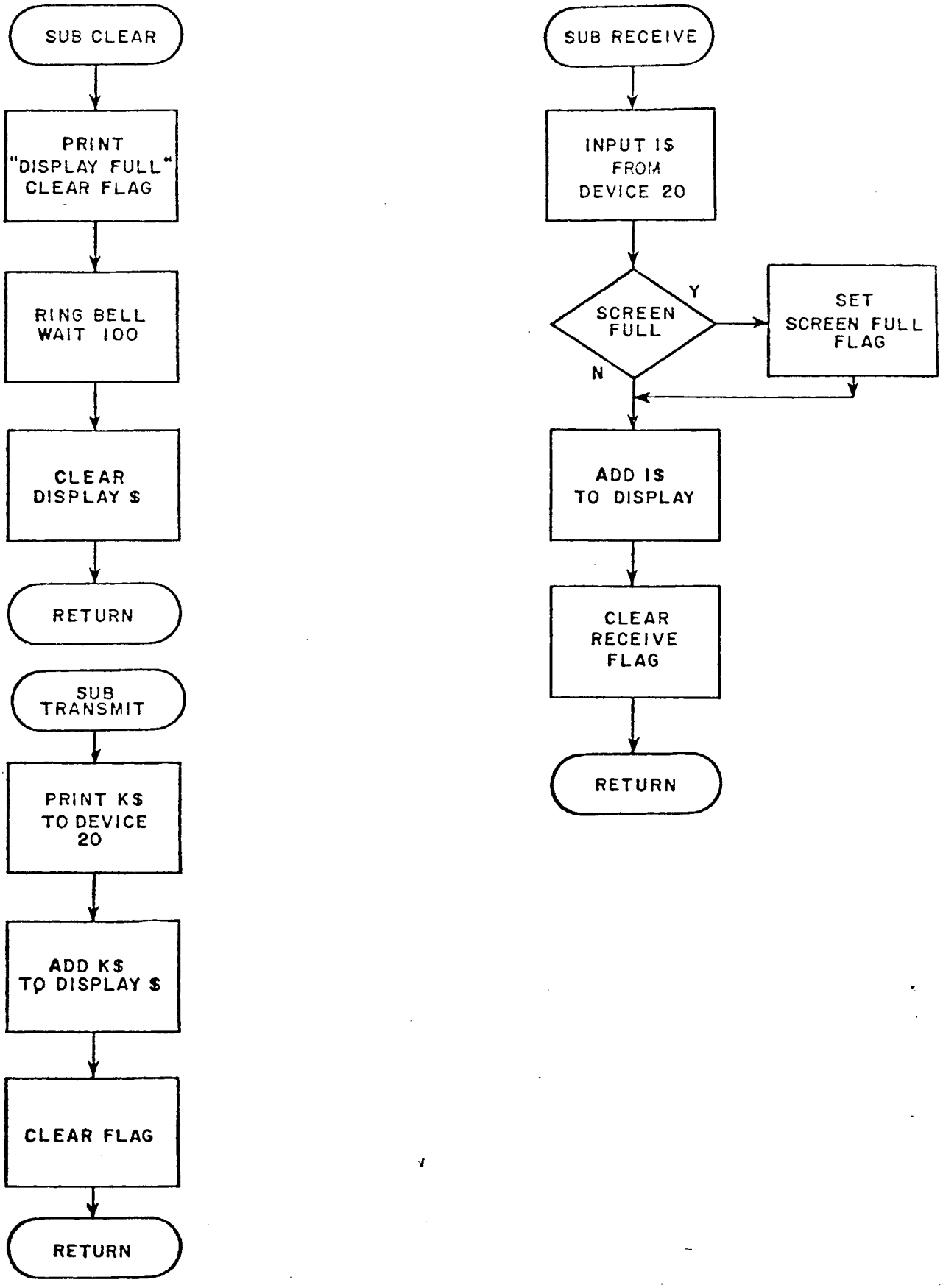


Figure 4. TEK 4051 SUBROUTINES

If the receive flag is set indicating that the 9835 has a message pending, the executive releases control to the receive subroutine (See Figure 4). The receive subroutine inputs the message from the 488 bus, checks for a screen full condition, and adds the message to the display string. The receive flag is reset to 0 and execution returns to the executive loop which goes on to perform a new display.

The reader should realize that this set of programs was developed very rapidly to demonstrate a capability. A few improvements are suggested here for future consideration.

1). Since the Tektronix special function key is an unconditional and overriding branch in program execution (It will even interrupt an input statement) it would be necessary to turn off the key using the "set no key" statement whenever the processor is engaged in any operation other than the executive loops.

2). The input of a message would be better accomplished on the occasion of the interrupt rather than returning to the executive loop first. The interrupt routine would not only set the appropriate flag but also bring in the message for further processing at a scheduled time.

3). An improvement in response to mesages both transmitted and received on the Tektronix could be gained by moving the flag checking code to the beginning of the executive loop.

5.2 HP-9835

Like the Tektronix 4051, the main program for the HP-9835 (See Figure 5) initializes variables and sets the interrupt conditions before beginning the executive loop. The loop is very simple, in this case only displaying and clearing the "MAIN PROGRAM EXECUTING" message. All other conditions are handled by interrupt.

When a key is pressed an interrupt is generated which branches program execution to the key service routine (See Figure 6). This routine adds the key stroke to the message string, displays the message at the top of the screen and as long as the (execute) key has not been pressed returns to program execution where the interrupt occurred. If the key pressed was the (execute) key then the SRQ (service request) line is set true before returning to the point of interruption.

When the Tektronix 4051 receives an SRQ true signal, the HP-9835 is addressed as a talker. When the HP-9835 is addressed as a talker or a listener, an interrupt is generated which results in program execution branching to the Teksevice routine (See Figure 6). This routine examines the status of the interface card to determine the reason for the interrupt. Program execution then branches to either the input subroutine or output subroutine depending on the whether the HP-9835 was addressed as a listener or a talker. After the appropriate routine has been executed and control returns to the Tekservice routine, a card enable statement is executed and program execution returns to the point of interruption.

When the output routine is called the message waiting to be transmitted is sent on the 488 bus and received by the 4051. The message is printed on the screen and the transmit string is cleared. The subroutine then checks for a full screen. If the screen is full, the clear screen routine is called and then in either case execution returns to the calling program (See Figure 7).

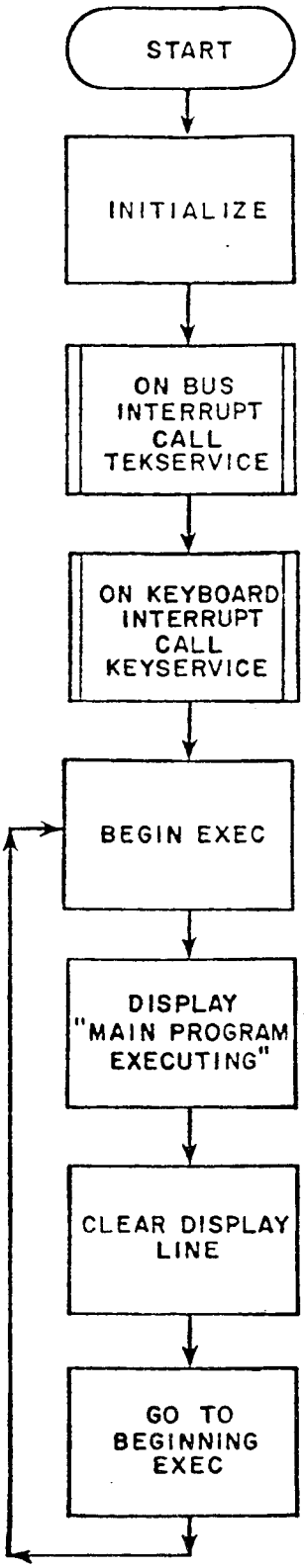


Figure 5. HP 9835 MAIN PROGRAM

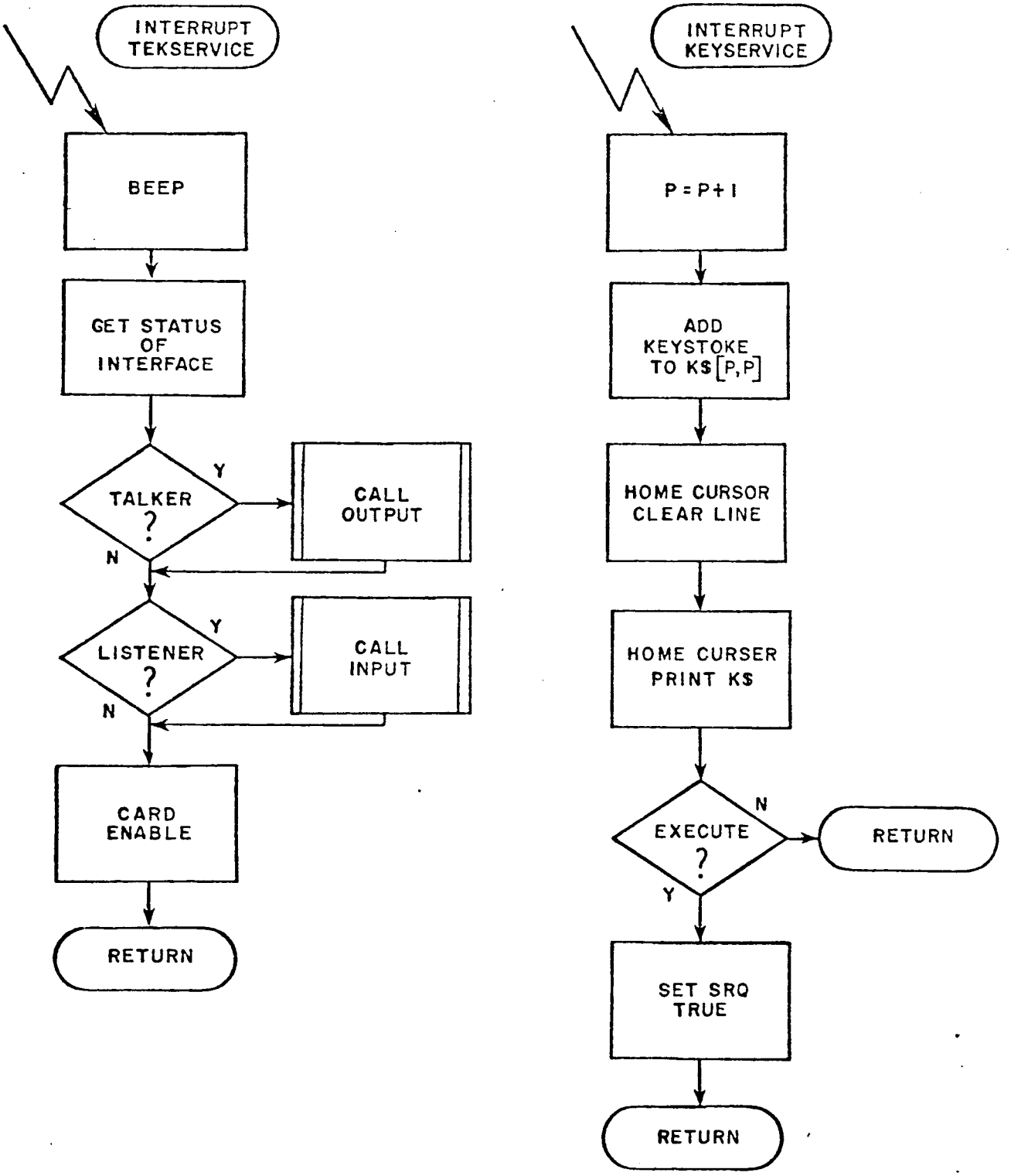


Figure 6. HP 9835 INTERRUPT SERVICE ROUTINES

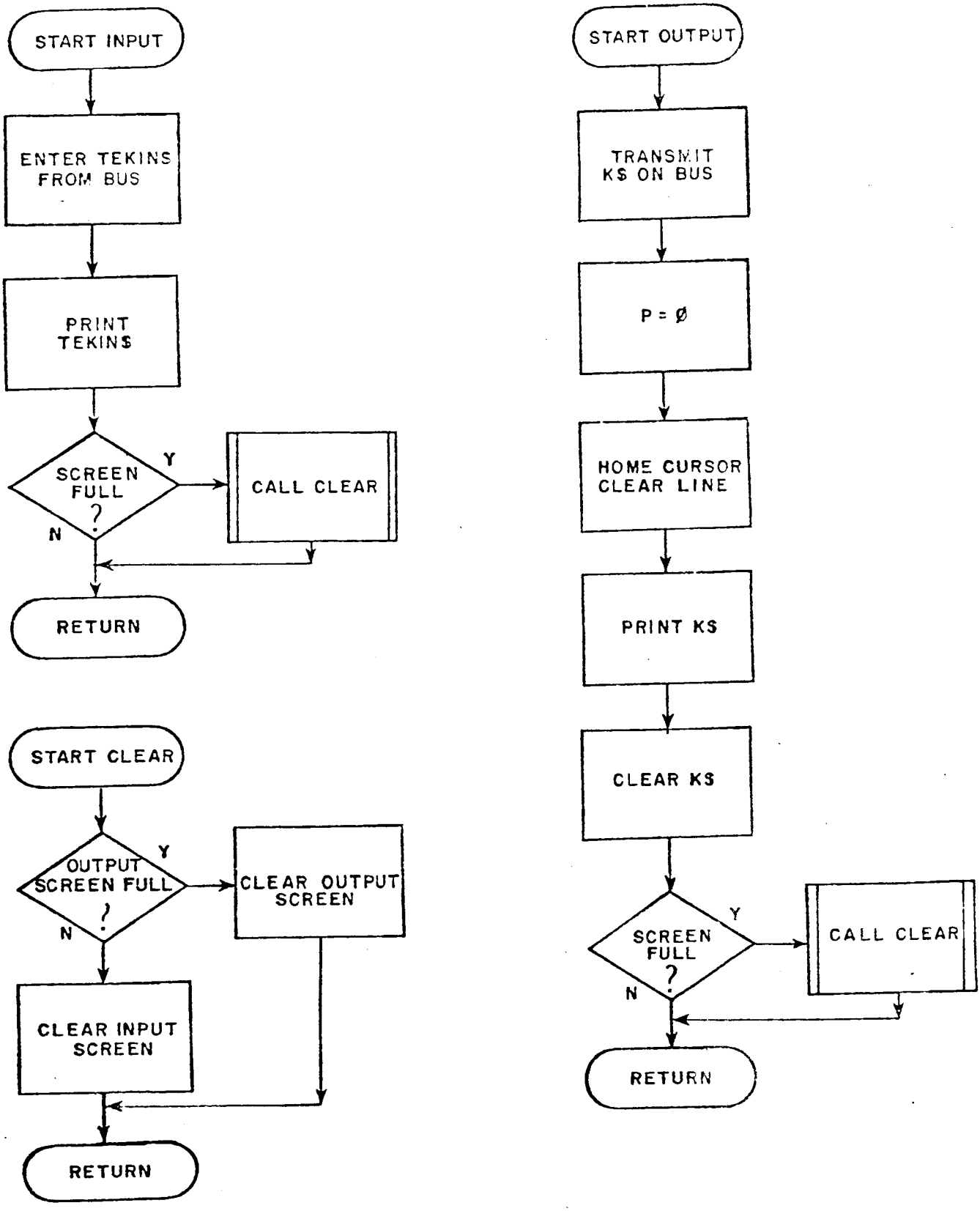


Figure 7. HP 9835 SUBROUTINES

When the input routine is called, the message from the Tektronix 4051 is input and printed on the screen. A screen full check is done and if full the clear screen subroutine is called. Execution then returns to the calling program, Tekservice (See Figure 7).

When the clear screen subroutine is called it first checks to see which screen is full and then clears that screen. Execution then returns to the calling program.

One possible enhancement for the HP-9835 software is worth noting. The keyboard interrupt should be disabled using the "off KBD" statement after the (execute) key signals the desire to transmit a message.

The interrupt should then be reenabled after the message has been transmitted. As the program stands a new message should not be entered until a beep is heard after pressing (execute) on the Hewlett-Packard 9835.

6. CONCLUSIONS

This evaluation has demonstrated the capability and feasibility of connecting the HP-9835 and the Tektronix 4051 using the IEEE-488 standard interface. The interrupt capabilities of both machines have been explored and utilized in a workable system. Based on the success of this evalution and the knowledge of each machine's structure, there is little doubt that it is possible to develop a communicating system using the 488 bus.

The capabilities are extant in both machines for programmable translation of input and output data streams. It should therefore be possible to receive a program from the Tektronix and store it on the Hewlett-Packard disk, retrieve it, and transmit it in a useable form back to the Tektronix.

This evaluation has indicated that a greatly improved ECLIPS could be developed using the Tektronix 4054 and Hewlett-Packard 9845 with the standard IEEE-488 interface used as a means of system communications. By using interrupts, the HP-9845 could insure that neither the user nor the Tektronix 4054 could lock out information being entered by the communications buffer. The architecture of the Tektronix 4054 does, however, allow the user to tie up the system which is acceptable since the Hewlett-Packard can hold any pending data until the Tektronix is free to process again.

The possibility of connecting computers on the 488 bus means that up to 14 HP computers could be tied together with a Tektronix controller (according to the standard). Only one Tektronix computer may be connected to any one bus since only one controller may be on a bus and the Tektronix computer does not have a feature which allows it to become a device instead of a controller. However, a system may be put together with a number of IEEE-488 buses (up to 14 using an HP-9845) with a Tektronix connected to each separate bus. Figure 8 shows a possible configuration where an HP computer coordinates a group of IEEE-488 buses which each have a Tektronix controller. With the advent of off-the-shelf fiber optics for the IEEE-488 link available from Hewlett-Packard, systems may be tied together over long distances using this high speed interface. Figure 9 shows a possible next generation ECLIPS suite with fiber optic links between three systems.

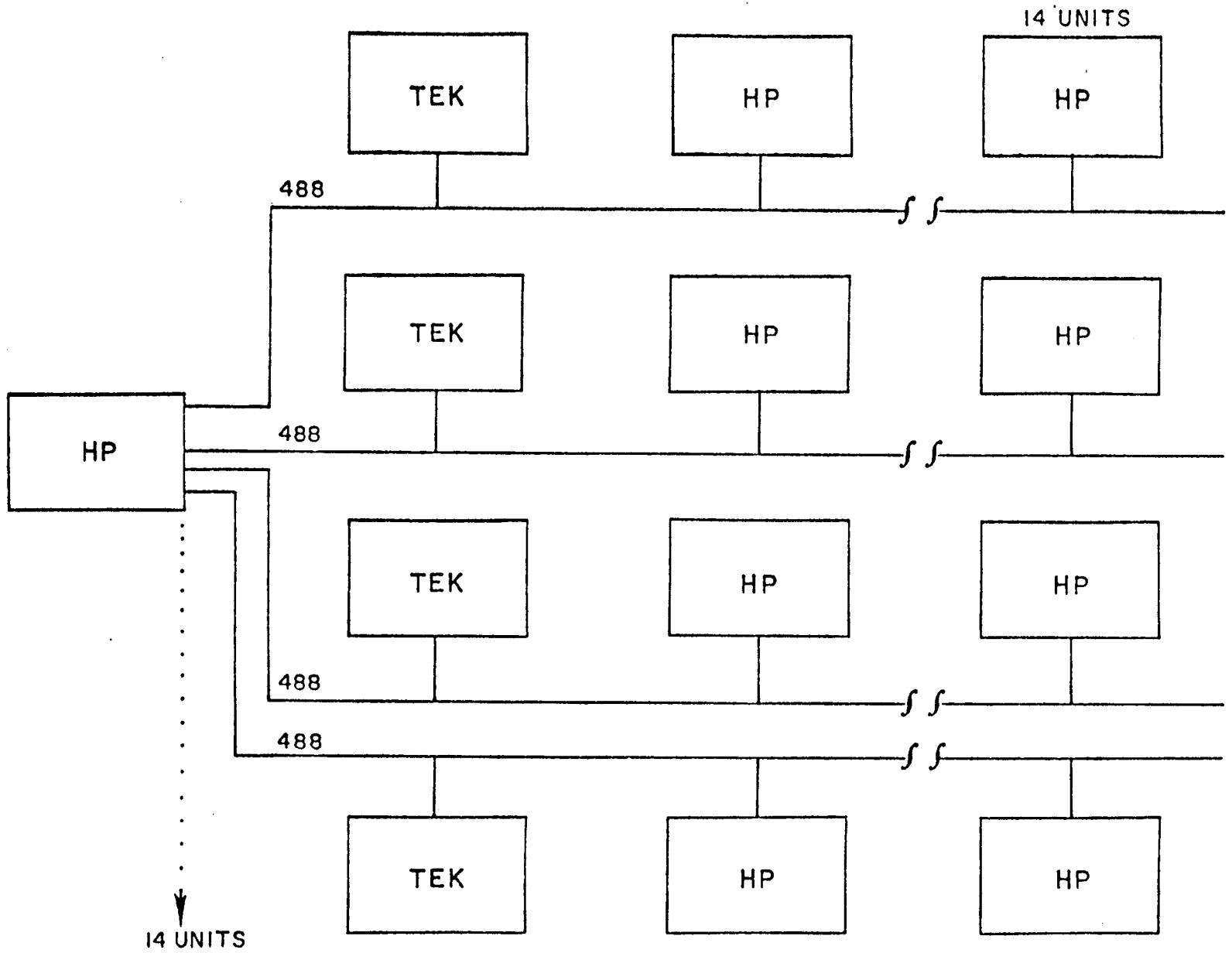

14 UNITS
TEK
HP
HP
488
TEK
HP
HP
HP
488
TEK
HP
HP
488
488
TEK
HP
HP
14 UNITS

Figure 8.

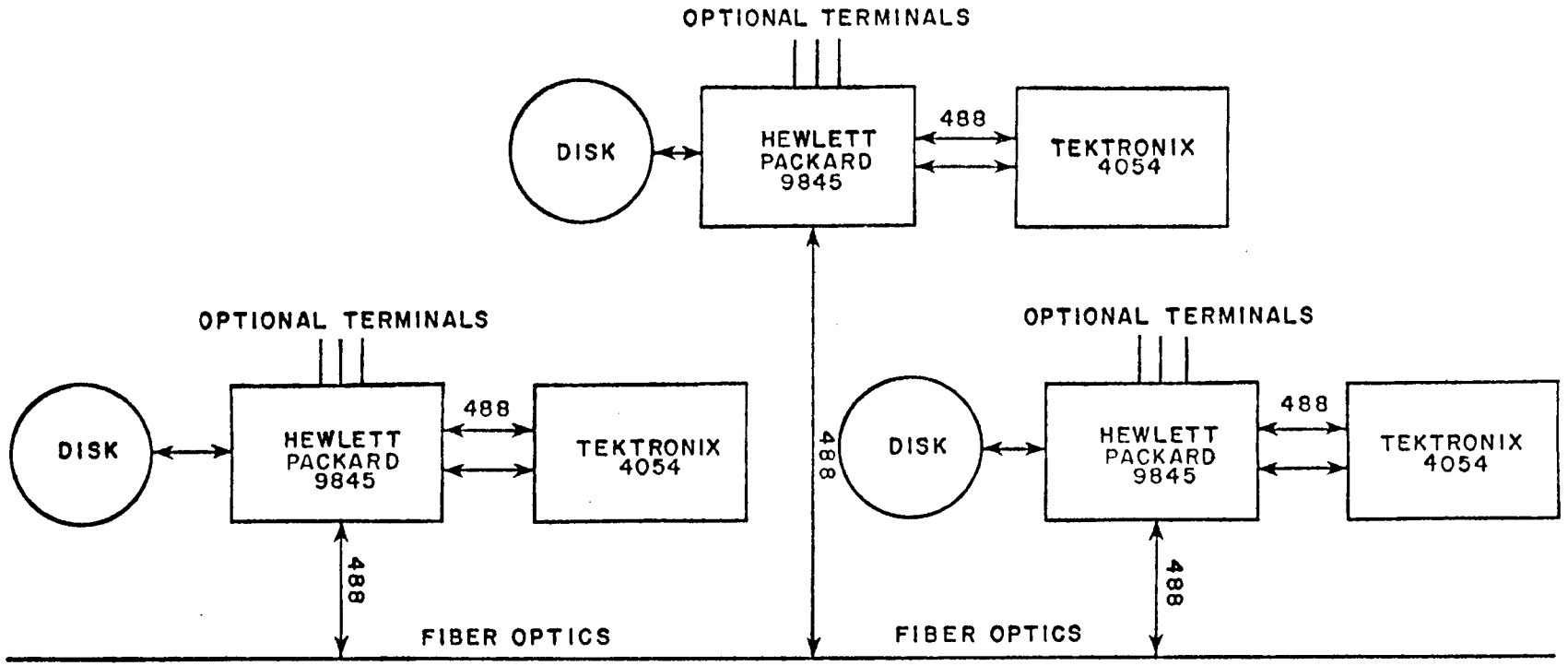

OPTIONAL TERMINALS
DISK
HEWLETT PACKARD 9845
488
TEKTRONIX 4054
OPTIONAL TERMINALS
DISK
HEWLETT PACKARD 9845
488
TEKTRONIX 4054
488
FIBER OPTICS
OPTIONAL TERMINALS
DISK
HEWLETT PACKARD 9845
488
TEKTRONIX 4054
488
FIBER OPTICS

Figure 9

To sum up, the IEEE-488 bus holds great potential for multi-computer systems consisting of off-the-shelf desktop computer technology. Any computer which supports the IEEE-488 interface is a candidate for inclusion into a system.

APPENDIX

```
3 GO TO 100
4 GOSUB 360
100 REM 9835- 4051 COMS 488
110 INIT
120 SET KEY
130 ON SRQ THEN 470
140 DIM A$(288),B$(288),I$(80),E$(80)
142 F=0
144 T=0
146 R=0
150 C$="DISPLAY FULL - MAKE COPY - PRESS KEY TO CONT"
155 E$=" "
160 GOSUB 510
170 PRINT "↥"
171 FOR I=1 TO 5
180 PRINT "MAIN PROGRAM EXECUTING"
190 NEXT I
200 MOVE 0,76
210 PRINT A$
220 MOVE 0,45
230 PRINT B$
240 IF F<>1 THEN 300
250 MOVE 0,11
260 PRINT C$
262 PRINT "G"
264 FOR X=1 TO 100
266 NEXT X
270 F=0
280 REMUT C
290 GOSUB 510
300 IF T<>1 THEN 320
310 GOSUB 420
320 IF R<>1 THEN 335
330 GOSUB 560
```

```
335 FOR I=1 TO 500
336 NEXT I
340 PAGE
350 GO TO 170
360 REM SUB KEY INTERUPT
361 MOVE 0,9
370 INPUT K$
380 T=1
390 IF LEN(A$)+LEN(K$)<288 THEN 410
400 F=1
410 GO TO 170
420 REM SUB TRANSMIT
430 PRINT @20:K$
440 A$=A$&K$
450 T=0
460 RETURN
470 REM SRQ INTERUPT
480 POLL M,W;20
490 R=1
500 RETURN
510 REM SUB STRING CLEAR
520 A$="TRANSMITTED TO 9835__"
530 B$="RECEIVED FROM 9835__"&E$
540 E$=" "
550 RETURN
560 REM SUB RECEIVE
570 INPUT @20:I$
580 IF LEN(B$)+LEN(I$)<288 THEN 630
590 F=1
600 E$=I$
610 R=0
620 RETURN
630 B$=B$&I$
640 R=0
650 RETURN
```

```
660 END
```

```
10    ! 9835-4051 CONS 488
20    ON INT #7,15 CALL Tekservice
30    CONTROL MASK 7;"0010000"
40    CARD ENABLE 7
50    COM C,E,I,L,K$[80],D,Tekins[300]
60 SET TIMEOUT 7;100
70    P=0
80    K$=""
90    C=E=5
100   L=2
110   D=11
120   ON KBD CALL Keyservice
130 DISP "MAIN PROGRAM EXECUTING"
140   REM
150   REM
160   REM
170   DISP CHR$(13)
180   GOTO 130
190 END
200   SUB Tekservice
210   BEEP
220   IF TIME OUT(7)>0 THEN 270
230   STATUS 7;Stat
240   IF BIT(Stat,4) THEN CALL Input
250 IF BIT(Stat,5) THEN CALL Output
260   GOTO 290
270 DISP "TIMEOUT - MESSAGE NOT SENT"
280   STOP
290   CARD ENABLE 7
300   SUBEND
310 SUB Clear(S)
350   IF S<1 THEN 430
```

```
360 FOR I=2& TO 10
370  PRINT USING "#,K";CHR$(27)&"H"
380  PRINT LIN(I);SPA(80)
390  NEXT I
400  L=2
420  SUBEXIT
430  FOR I=11 TO 20
440  PRINT USING "#,K";CHR$(27)&"H"
450  PRINT LIN(I);SPA(80)
460  NEXT I
470  D=11
490  SUBEND
500  SUB Input
510  COM C,E,L,D,K$[80],P,Tekin$[80]
520  ENTER 7;Tekin$
530  PRINT USING "#,K";CHR$(27)&"H"
550  PRINT LIN(L);Tekin$
580  L=L+1
600  IF L>10 THEN CALL Clear(1)
610  SUBEND
620  SUB Output
630  COM C,E,L,D,K$[80],P
640  OUTPUT 7;K$
650  P=0
660  DISP CHR$(12)
670  PRINT USING "#,K";CHR$(27)&"H"
671  PRINT SPA(80)
675  PRINT USING "#,K";CHR$(27)&"H"
680  PRINT LIN(D);K$
690  K$=""
700  D=D+1
```

```
710  IF I=20 THEN CALL Clear(0)
720  SUBEND
730  SUB Keyservice
740 CON C,F,L,U,K$(80),P
750  P=P+1
760  K$(P,P1)=KBD$
770  PRINT USING "#,K";CHR$(27)&"H"
780  PRINT SPA(80)
785  PRINT USING "#,K";CHR$(27)&"H"
790  PRINT K$
791  IF LEN(K$)>79 THEN 810
800  IF NUM(K$(P,P1))<>255 THEN SUBEXIT
810  REQUEST 7;64
820  SUBEND
```

TEKTRONIX

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
20mA Current Loop Interface	51/00-6118/0
ORIGINAL DATE 3/80 / REVISION DATE	MEMORY REQUIREMENT 8K
AUTHOR William M. Retallack, John H. Taylor, NOAA - Boulder, CO	PERIPHERALS Model 33 Teletype

ABSTRACT

Files: 1 ASCII Program
Statements: 52

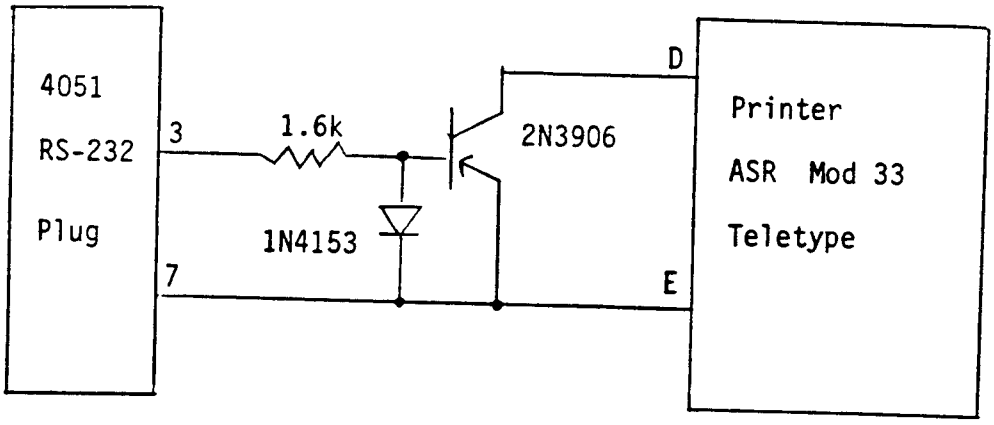
One of the many different tasks we have in data reduction is digitizing analog data and outputting long lists of X and Y values. To save on the cost of hard copy paper, a special interface was constructed. The analog data is digitized, scaled and the output is listed on a Model 33 Teletype. It has been in use for over three years and hasn't dropped a bit yet.

The interface connects through the RS-232 plug. Complete schematics are included along with a short digitizer and output program.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0

RS-232-C to 20mA Current Loop Interface



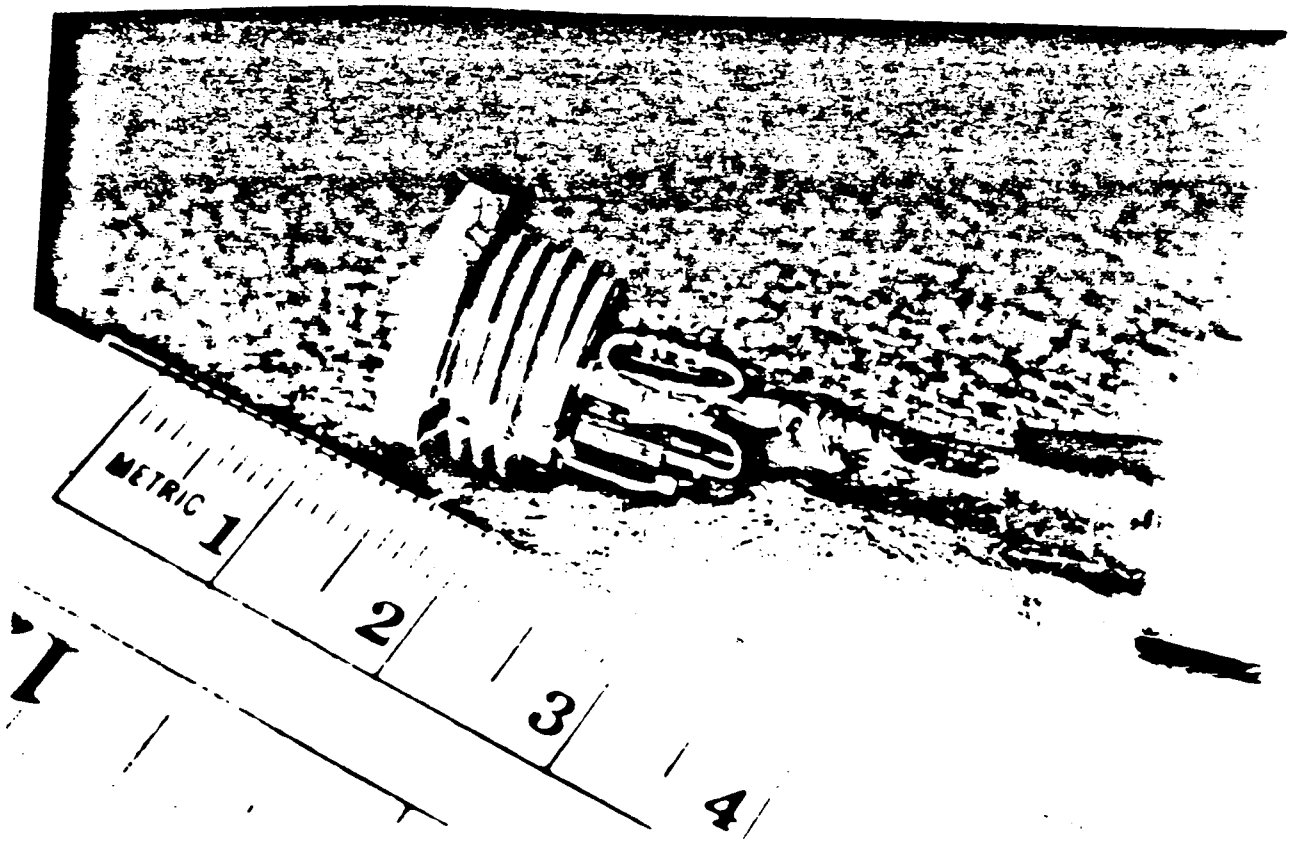
On back of plug

1. Jumper pins 4, 5 and 20
2. Jumper pins 11, 12 and 19

Cost:

1.	2N3906	0.15
2.	1N4153	0.10
3.	1.6k	0.08
4.	Plug & cable	3.00
		$3.33

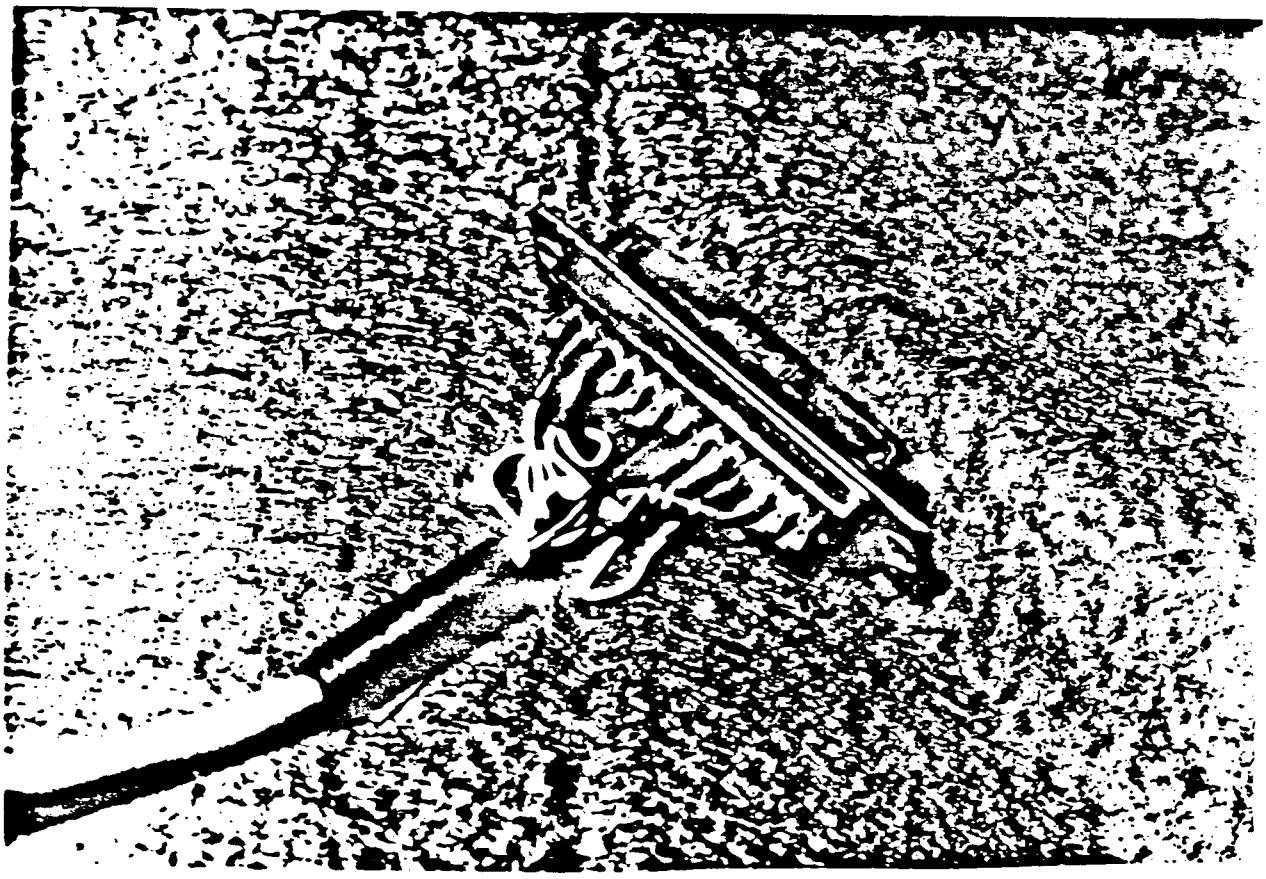
TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0



Connector to Model 33 Teletype

(see the actual photo in the 4050 Series Applications Library catalog)

TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0



Connector to 4050 Series RS-232-C Data Communications Interface

(see the actual photo in the 4050 Series Applications Library catalog)

TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0

Variables	Used to Store	Type
D	Output address	Simple
M(1)	MIN-X value of data	Array
M(2)	MAX-X " " "	"
M(3)	MIN-Y " " "	"
M(4)	MAX-Y " " "	"
M(5)	)	Array
M(6)	)	"
A	) skew correction	Simple
B	)	"
C	) and scaling	"
RØ,R1,R2,R3	)	"
I	Loop Index	Simple
X,Y	coordinates	"
Z$	Status of Tablet Cursor	String

EQUIPMENT REQUIRED

4050 System
Tablet with four-button cursor, use the white button (status = 1) to digitize
Model 33 Teletype
Specially constructed interface

TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0

Operating Instructions

Set your tablet controller to POINT mode.

Enter the data from the keyboard as requested.

Lay your graph or plot on the tablet.

Move the four-button cursor to the point of origin on your plot (the location of the minimum X and Y data, (see next page)

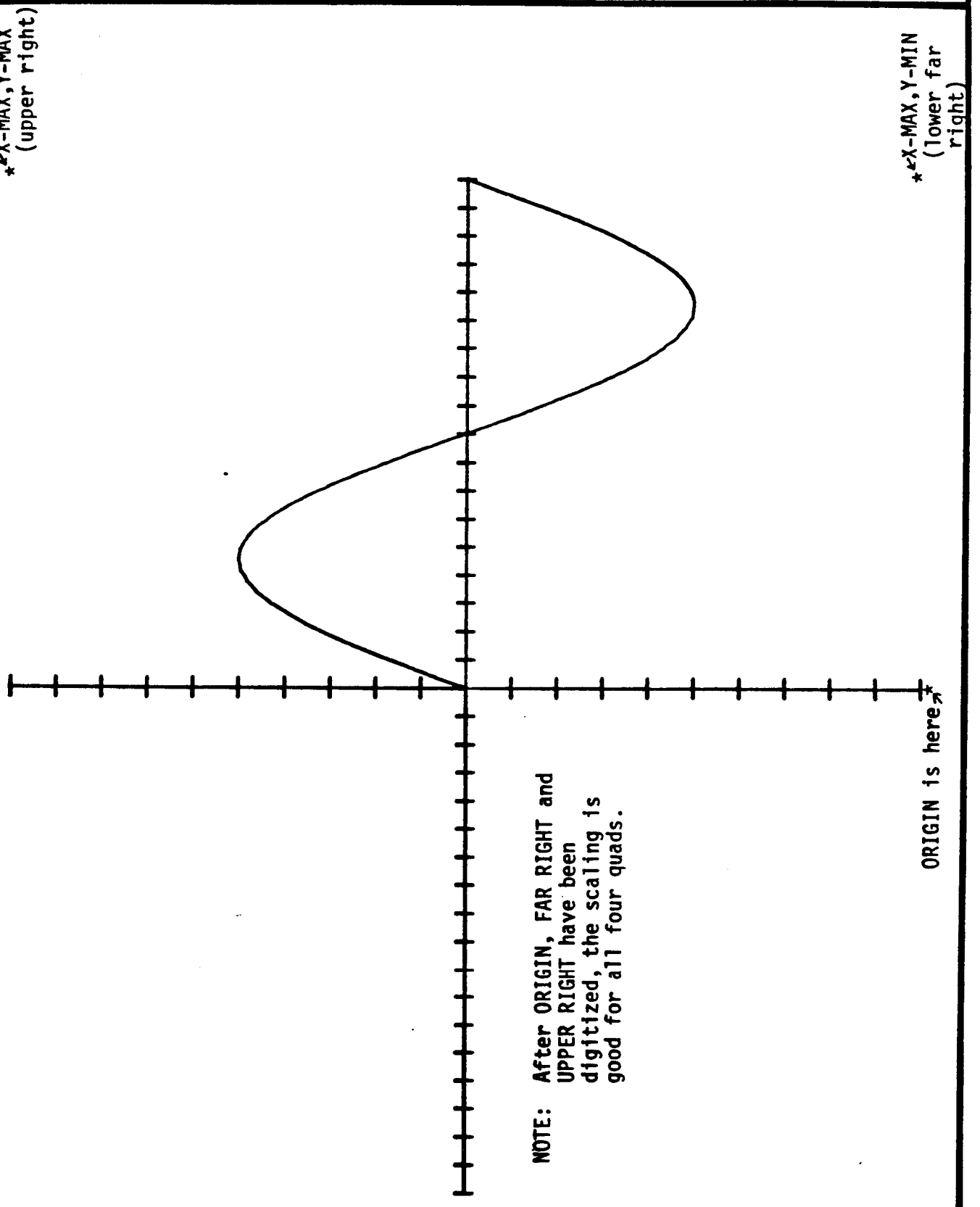
While holding in the ORIGIN button, digitize this point.

Release the ORIGIN button and digitize the lower far right of your plot; now digitize the upper far right.

You are now ready to digitize the plot and have the data transferred to the Model 33 Teletype. Press Button 2 to end the session.

This is a very simple program and could certainly be expanded to operate in STREAM mode, etc.

TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0


*X-MAX,Y-MAX (upper right)
*X-MAX,Y-MIN (lower far right)
ORIGIN is here*
NOTE: After ORIGIN, FAR RIGHT and UPPER RIGHT have been digitized, the scaling is good for all four quads.

TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0

```
100 INIT
110 PAGE
120 PRINT "This program will digitize a graph or magnetogram "
130 PRINT "and then list the data out to the Mod. 33"
140 DIM M(6)
150 PRINT "JJEnter X-MIN value of data ";
160 INPUT M(1)
170 PRINT "JEnter X-MAX value of data ";
180 INPUT M(2)
190 PRINT "JEnter the Y-MIN value of data ";
200 INPUT M(3)
210 PRINT "JEnter the Y-MAX value of data ";
220 INPUT M(4)

230 REM---Find the range of X and Y

240 M(2)=M(2)-M(1)
250 M(4)=M(4)-M(3)
260 PRINT "JRange of X= ";M(2);" Range of Y= ";M(4)
270 PRINT "JSet the ORIGIN of the board at X-MIN,Y-MIN"
280 INPUT @8:M(5),M(6),Z$
290 PRINT "JGGGORIGIN=";M(5);",";M(6)
300 PRINT "JGNow digitize the far right of data (X-MAX,Y-Min)"
310 INPUT @8:M(5),M(6),Z$

320 REM --- Get the rotation information

330 A=ATN(M(6)/M(5))
340 B=SIN(A)
350 C=COS(A)

360 REM --- Get X-MAX and Y-MAX

370 PRINT "JGNow digitize the upper right (X-MAX,Y-MAX)"
380 INPUT @8:X,Y,Z$
390 PRINT "GG"
```

TITLE	ABSTRACT NO:
20mA Current Loop Interface	51/00-6118/0

```
400 REM --- Calculate the rotation

410 R0=X*C+Y*B
420 R1=Y*C-X*B
430 PRINT "Now go and sample the data "

440 REM --- MAIN DIGITIZING LOOP ---

450 PRINT @D:"L"
460 PRI @D:"     X          Y          X          Y          X          Y"
470 FOR I=1 TO 6 STEP 2
480     INPUT @8:X,Y,Z$
490     Z=VAL(Z$)
500     GO TO Z OF 510,610
510     R2=X*C+Y*B
520     R3=Y*C-X*B
530     X=M(1)+M(2)*R2/R0
540     Y=M(3)+M(4)*R3/R1
550     PRINT @D: USING 560:X,Y
560     IMAGE 2(6D.3D),S
570 NEXT I
580 PRINT @D:"J"
590 GO TO 470
600 PRINT @D:"L"
610 END
```

TEKTRONIX

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
Leroy Character Generator	51/00-8012/1
ORIGINAL DATE 6/77 / REVISION DATE 6/80	MEMORY REQUIREMENT 13597 Bytes (Full Version)
AUTHOR Mark Mehall	PERIPHERALS 4662 (Optional)

ABSTRACT

This set of routines draws English and Greek upper and lower case letters, numbers, special characters, and centered symbols on a 4050 or 4662. The characters and symbols can be any size, at any angle and may be positioned at any location on the screen or plotter surface. Since routines do not interfere with the user's window and viewport, they can be used with existing programs for annotation. Labels can be placed on axes, points can be indicated, and legends can be added to graphs. There are 211 different symbols available. The routines also offer subscripts and superscripts, underlining, *italics*, tab settings, backspace and carriage return functions. The characters are similar to the Leroy style.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

DESCRIPTION

There are three routines in this package and one data file. The first routine (GOSUB 9000) initializes the tables used by the program from the data file. This routine should be used only once during each program. The second routine (GOSUB 9100) does the actual character generation. The routine requires three variables: C$ – the string of ASCII characters to be drawn, S – the size of the characters in Graphic Display Units (GDUs), and A – the angle (in degrees) of counterclockwise rotation. The third routine (GOSUB 9200) is an entry point to allow some special features to be used. It will be discussed later.

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

TAPE LAYOUT

There are two files associated with this set of routines. The first is an ASCII file containing the three subroutines needed for the character generation. The second is a binary file containing the character arrays N$, D$, and M$. When GOSUB 9000 is executed, it finds the binary file (statement 9014) and reads the arrays. If the binary file is not the second file on the tape, statement 9014 should be changed to indicate the correct file number.

SUGGESTED USAGE

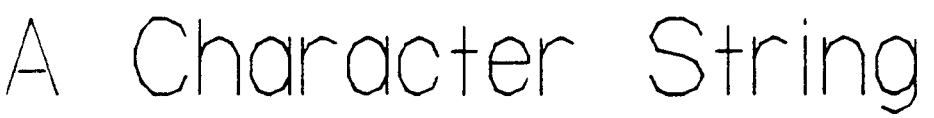
After a program has been written to use the character generator routines, make a REMARK statement at line 9000, insert the tape with the character generator and FIND the file which has the three routines. The routines can now be added using APPEND 9000 The user program should execute a call to the initializing routine (GOSUB 9000) before the character generator can be used. Alternatively, the user can use RUN 9000 to initialize the tables without having GOSUB 9000 in his program. The initializing routine should only be executed once for each program.

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

EXAMPLE #1 PROGRAM

```
100 REM INITIALIZE THE CHARACTER GENERATOR
110 GOSUB 9000
120 REM SET ANGLE TO 0
130 A=0
140 REM SET CHARACTER STRING
150 C$="A Character String"
160 REM SET THE SCALE TO 4 (GDUs)
170 S=4
180 REM MOVE TO THE BEGINNING LOCATION
190 MOVE 65,50
200 REM EXECUTE THE ROUTINE
210 GOSUB 9100
220 END
```

EXAMPLE #1 RESULT

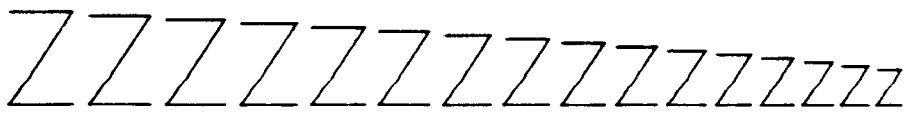


TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

EXAMPLE #2 PROGRAM

```
100 REM SET ANGLE TO 0
110 A=0
120 REM SET CHARACTER STRING TO BE "Z"
130 C$="Z"
140 REM MOVE TO THE BEGINNING LOCATION
150 MOVE 65,50
160 REM SET THE SIZE OF THE STRING
170 FOR S=5 TO 2 STEP -0.2
180 REM EXECUTE THE ROUTINE
190 GOSUB 9100
200 NEXT S
210 END
```

EXAMPLE #2 RESULT



TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

EXAMPLE #3 PROGRAM

```
100 REM SET CHARACTER STRING
110 C$="  Symbol Generator"
120 REM SET THE SCALE TO 2.5 (GDUs)
130 S=2.5
140 REM SET THE ANGLE
150 FOR A=0 TO 330 STEP 30
160 REM MOVE TO THE BEGINNING LOCATION
170 MOVE 65,50
180 REM EXECUTE THE ROUTINE
190 GOSUB 9100
200 NEXT A
210 END
```

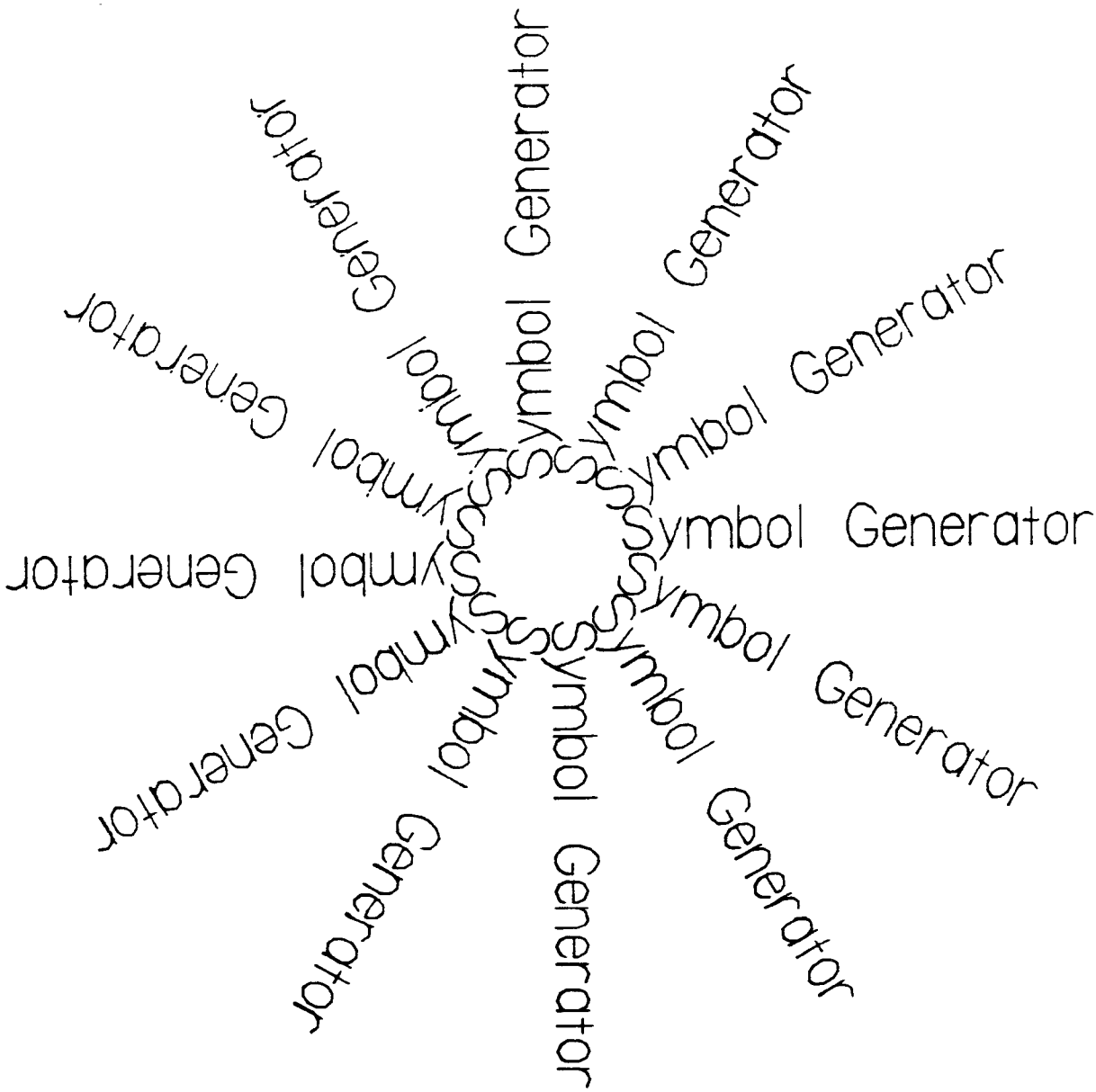
TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

EXAMPLE #3 RESULT



TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

CHARACTERS AVAILABLE

There are 256 characters in this set of routines. They are divided into two groups (English and Greek). The printable ASCII characters (decimal 32 through 126) may be obtained directly from the 4050 keyboard. The ASCII control codes (decimal 0 through 31) have been defined to produce additional characters (see tables). Decimal 0 through 15 of the English group are special symbols which are centered about the location of the beam (or pen) when the routine is entered. The control code characters can be obtained by using the CTRL key on the 4050. For example C$="A" (CTRL A) produces ⊙

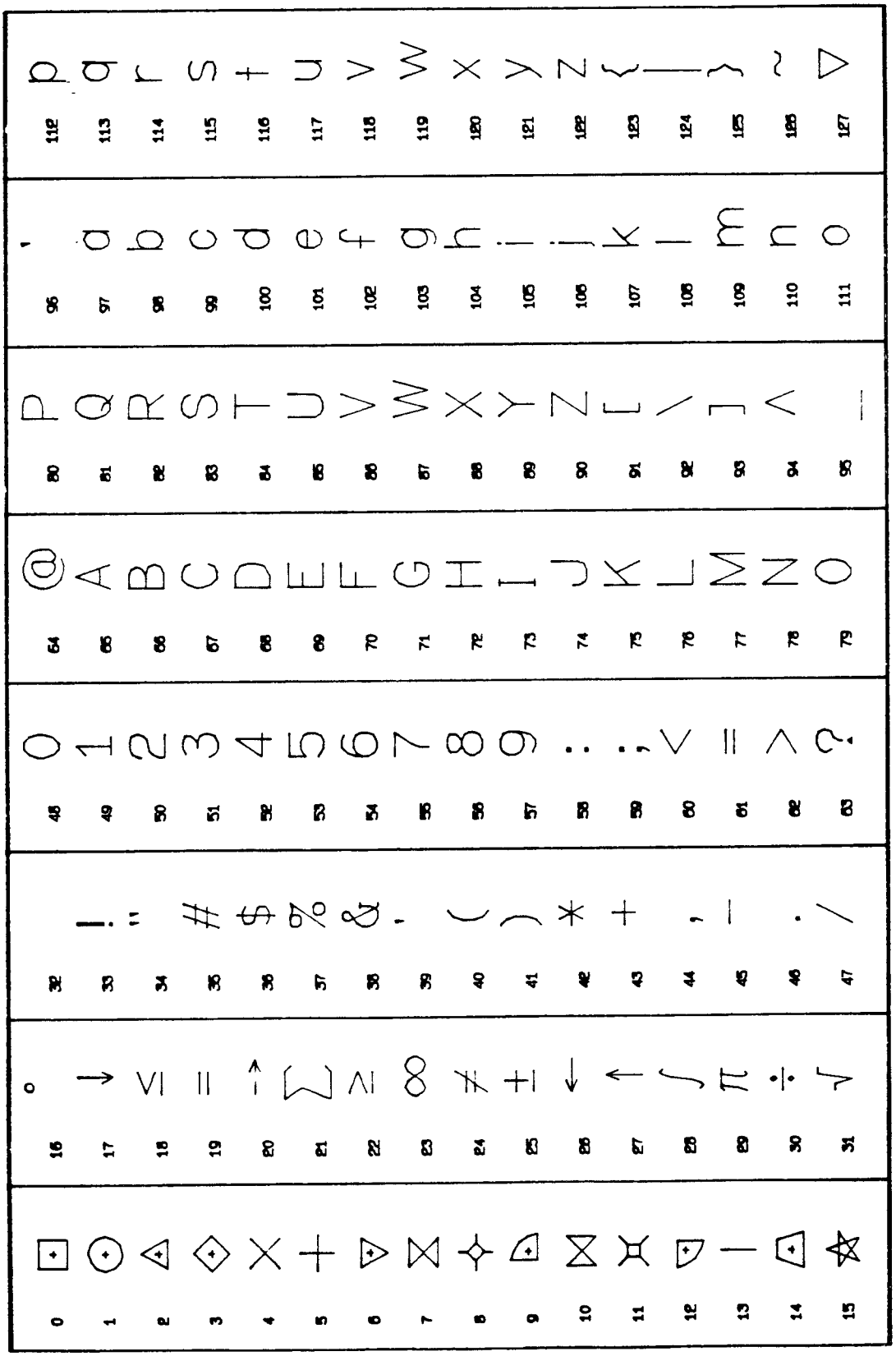
There are two characters which cannot be obtained directly from the keyboard. Both decimal 13 and 127 must be defined using the CHR function (e.g. C$=CHR(127)).

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1



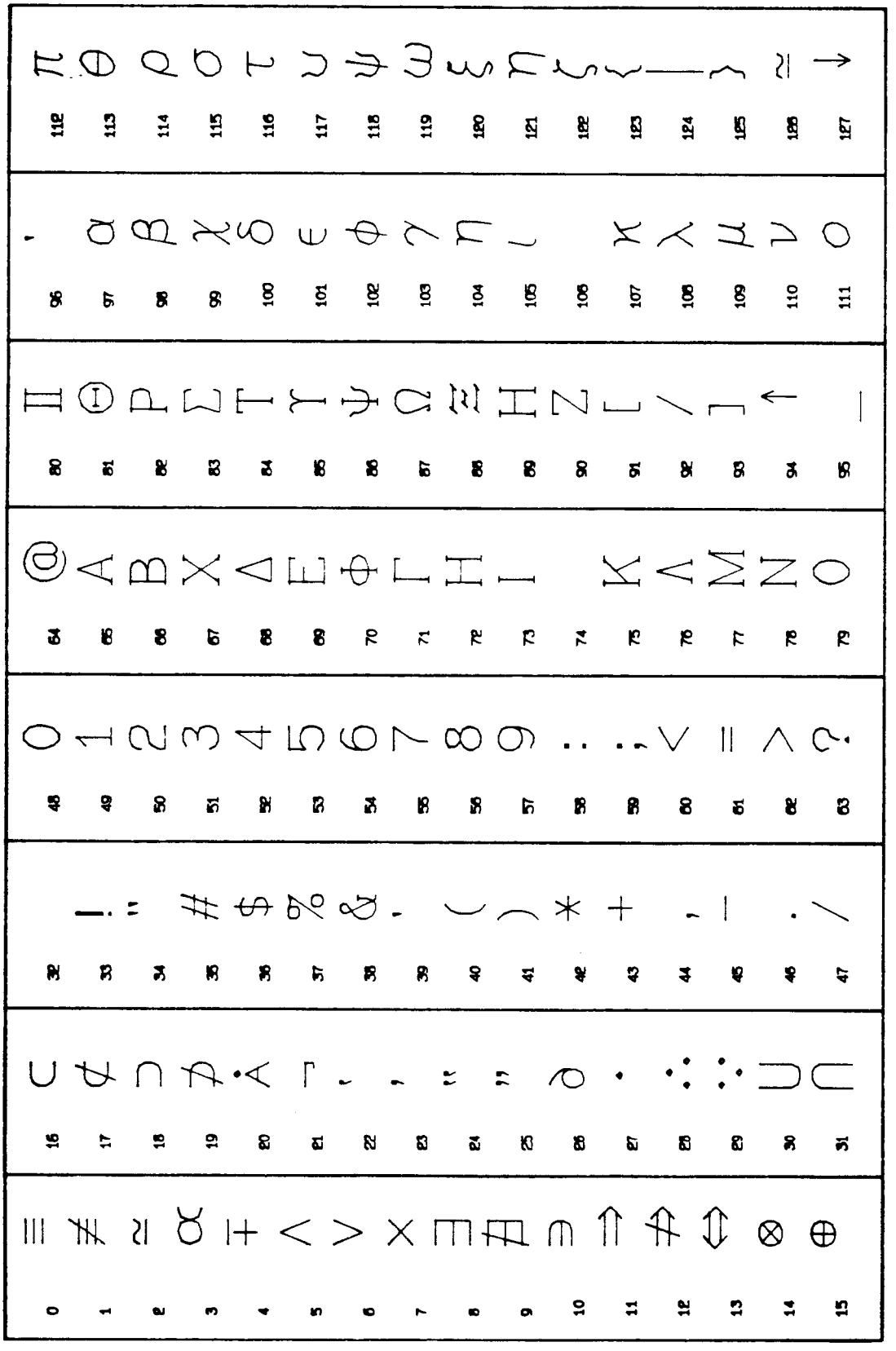
English Characters

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1



Greek Characters

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

FEATURES

The control character | is used to convey special instructions according to the character which follows the |. For example, |7 means to enter the subscript mode. If a | is followed by a character which is not defined as a code, it will be ignored. A single | is drawn if two of them appear successively. Control characters may appear anywhere in C$.

English and Greek

The characters |3 invoke the English group of characters. This is also the default group used if none is specified. The Greek characters are enabled by using |4. For example: the string C$="tan(|4a+b)" produces: tan(α+β) Note that the Greek group will be used until the English set is enabled with a |3.

Superscripts and subscripts

If the characters |5, |6, |7, or |8, are encountered, an instruction regarding superscripts and subscripts is executed:

|5 means to enter the superscript mode;
|6 means to leave the superscript mode;
|7 means to enter the subscript mode;
|8 means to leave the subscript mode.

Superscripts may have subscripts or superscripts, etc. For example the string C$="e|5x|7k|8|52(n-1)|6|6dx|7k|8" produces the result:

$$e^{x_k^{2(n-1)}dx_k}$$

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

Establishing a reference point and returning to it

If the characters |9 are encountered, the coordinates of the next available position are stored for future reference. If the characters |A are encountered, the drawing will continue at the position stored in the last reference to |9. If |9 has not used, the reference point is assumed to be the starting location used when GOSUB 9100 was invoked. Since GOSUB 9100 resets the reference point, GOSUB 9200 should be used with |9.

Tab

A single "tab setting" is provided. The position of the tab setting is established (or revised) by using the control characters |B in C$ when at the desired location. Any subsequent use of the control characters |C will cause the following characters to be drawn at a point directly below the point at which |B was invoked. If |B has not been used, it is assumed that the tab was set at the starting location when GOSUB 9100 was invoked. The tab (|C) is usually used with multiple calls to carriage return (see below). The "tab set" position is preserved from one call to GOSUB 9100 or GOSUB 9200 to another.

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

Underlining

The underscoring of a word or a series of words is accomplished by the use of the code |9 to establish the beginning of the underscored material and the code |D to establish the end of the underscored material. For example the string C$="The |9title|D" produces the result:

The title

If |D is invoked without |9 ever having been used, underlining will start from the starting location when GOSUB 9100 was executed.

Backspace

If the characters |E are encountered, drawing will continue *starting at the beginning of the last drawn character.* Backspace may not be called meaningfully more than once in succession, since the only pertinent information stored is the beginning location of the last plotted character.

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

Carriage return

The control characters |F are used when drawing is to be resumed at a point located at 1.5*S below the beginning point of the previous execution of GOSUB 9100. Carriage returns may be used successively.

Italics

When a group of letters is to be *italicized*, the control characters |G are placed before the group and the control characters |H after the group, in C$. For example, the string C$="|Gtime|H, T" produces the result:

time, T

Italics are "carried over" from one call to GOSUB 9100 (or GOSUB 9200) to the next.
For example, the series of statements:

```
C$="|G"
GOSUB 9100
C$=STR(PI)
GOSUB 9200
C$="|H r|52"
GOSUB 9200
```

produces the result:

3.14159265359 r^2

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

MEMORY REQUIREMENTS

The full version takes 13597 bytes of storage. The following can be used to decrease the amount of memory required:

1. Delete all REM statements (except 9000, 9100 and 9200). This version uses 10367 bytes of memory.

2. Remove the special functions by deleting lines 9224 through 9376, and delete the Greek Characters by running the following program:

```
100 DIM N$(513),D$(1025),M$(5086)
110 FIND 2
120 READ @33:N$,D$,M$
130 DIM N$(256),D$(512),M$(2278)
140 FIND 2
150 WRITE @33:N$,D$,M$
160 CLOSE
170 END
```

   Also change line 9006 to:

```
9006 DIM N$(257),D$(513),M$(2279),E$(1),F$(1),G$(1)
```

   This version uses 5371 bytes of memory.

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

REVISION

This revision (6/80) included the following:

1. Greek letters and 33 new symbols added.
2. Added English (|3) and Greek (|4) codes.
3. Character #16, "P" the degree symbol (°), is no longer centered.
4. Increased character spacing for improved readability.

REMARKS

This set of routines is based on a program developed at the University of Illinois by J. W. Phillips. Please send any comments or suggestions to:

Mark Mehall
Tektronix, Inc.
P.O. Box 500 63-425
Beaverton, OR 97077

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

PROGRAM LISTING

```
9000 REM THIS ROUTINE INITIALIZES THE TABLES USE BY THE
9002 REM CHARACTER GENERATOR. IT SHOULD BE EXECUTED (ONCE) BEFORE
9004 REM THE CHARACTER GENERATOR IS USED. (i.e. GOSUB 9000)

9006 DIM N$(513),D$(1025),M$(5087),E$(1),F$(1),G$(1)

9008 REM SET GRAPHIC DEVICE (DEFAULT IS SCREEN)

9010 U9=32

9012 REM FIND FILE WITH DATA ARRAYS AND INITIALIZE

9014 FIND 2
9016 READ @33:N$,D$,M$

9018 REM SET CODE FOR NO ITALICS

9020 I9=0

9022 REM SET CODE FOR ENGLISH CHARACTER SET

9024 J9=0
9026 RETURN

9100 REM CHARACTER GENERATOR ROUTINE
9102 REM VARIABLES USED -
9104 REM   S - SCALE OF THE CHARACTERS IN GRAPHIC DISPLAY UNITS (GDUs)
9106 REM   C$ - ASCII CHARACTER STRING TO BE PLOTTED
9108 REM   A     - ANGLE (IN DEGREES) OF COUNTERCLOCKWISE ROTATION
9110 REM LOCAL VARIABLES USED -
9112 REM   X4,Y4    - TAB REFERENCE ALSO UNDERLINE REFERENCE
9114 REM   X5,Y5    - GENERAL REFERENCE POINT
9116 REM   X6,Y6    - CURRENT LOCATION
9118 REM   X7,Y7    - CHARACTER MOVES
9120 REM   X8,Y8    - STARTING CHARACTER STRING LOCATION USED FOR CR
9122 REM   X9,Y9    - WORK VARIABLES
9124 REM   S9,C9  - SIN(A) AND COS(A)
9126 REM   E$,F$,G$ - WORK STRINGS
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
9128 REM   N$   - TABLE OF MOVES PER CHARACTER
9130 REM   D$   - TABLE OF INDEX INTO M$ AND WIDTH
9132 REM   M$   - CHARACTER MOVES
9134 REM   I9   - ITALICS CODE (0=OFF 1=ON)
9136 REM   J9   - CHARACTER SET CODE (0=ENGLISH 1=GREEK)
9138 REM   L9,M9 - INDEX VARIABLES
9140 REM   P9   - DRAW CODE (1=MOVE 2=DRAW)
9142 REM   S8   - CORRECTED SCALE FACTOR
9144 REM   U9   - UNIT NUMBER OF GRAPHIC DEVICE (DEFAULT IS 32)
9146 REM   C8   - ASCII DECIMAL EQUIVALENT OF LETTER OF C$
9148 REM   W8   - CHARACTER WIDTH (IN GDUs) USED FOR BACKSPACE
9150 REM   N9,D9 - MISC. VARIABLES
9152 REM GET BEGINNING LOCATION

9154 INPUT @U9,24:X6,Y6

9156 REM SAVE STARTING LOCATION

9158 X8=X6
9160 Y8=Y6

9162 REM SET REFERENCE POINT

9164 X5=X6
9166 Y5=Y6

9200 REM ENTRY POINT FOR BYPASSING THE GIN COMMAND
9202 REM THE CHARACTERS WILL BE DRAWN AT THE END OF THE LAST STRING
9204 REM CORRECT SCALE FACTOR FOR GDUs

9206 S8=S*1.25
9208 SET DEGREES

9210 REM ALL ANGLES IN DEGREES

9212 S9=SIN(A)
9214 C9=COS(A)
9216 FOR L9=1 TO LEN(C$)

9218     REM GET A CHARACTER
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
9220    E$=SEG$(C$,L9,1)
9222    C8=ASC(E$)+J9*128

9224    REM CHECK FOR CONTROL CHARACTER

9226    IF E$()";" THEN 9378
9228    L9=L9+1
9230    E$=SEG$(C$,L9,1)
9232    C8=ASC(E$)

9234    REM CHECK FOR ANOTHER CONTROL CHARACTER, IF YES DRAW ONE

9236    IF C8=124 THEN 9378

9238    REM CHECK FOR 3 THROUGH 9

9240    IF C8<51 OR C8>57 THEN 9246
9242    GO TO C8-50 OF 9252,9258,9264,9274,9284,9294,9304

9244    REM CHECK FOR A THROUGH G

9246    IF C8<65 OR C8>72 THEN 9532
9248    GO TO C8-64 OF 9312,9320,9328,9336,9344,9352,9368,9374

9250    REM ENGLISH CHARACTER SET MODE (CODE 3)

9252    J9=0
9254    GO TO 9532

9256    REM GREEK CHARACTER SET MODE (CODE 4)

9258    J9=1
9260    GO TO 9532

9262    REM ENTER SUPERSCRIPT MODE (CODE 5)

9264    Y6=Y6+S8*0.3333*C9
9266    X6=X6-S8*0.3333*S9
9268    S8=S8*0.6
9270    GO TO 9530
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
9272    REM LEAVE SUPERSCRIPT MODE (CODE 6)

9274    S8=S8*1.6667
9276    Y6=Y6-S8*0.3333*C9
9278    X6=X6+S8*0.3333*S9
9280    GO TO 9530

9282    REM ENTER SUBSCRIPT MODE (CODE 7)

9284    Y6=Y6-S8*0.2667*C9
9286    X6=X6+S8*0.2667*S9
9288    S8=S8*0.6
9290    GO TO 9530

9292    REM LEAVE SUBSCRIPT MODE (CODE 8)

9294    S8=S8*1.6667
9296    Y6=Y6+S8*0.2667*C9
9298    X6=X6-S8*0.2667*S9
9300    GO TO 9530

9302    REM ESTABLISH REFERENCE POINT (CODE 9)

9304    X5=X6
9306    Y5=Y6
9308    GO TO 9532

9310    REM RETURN TO REFERENCE POINT (CODE A)

9312    X6=X5
9314    Y6=Y5
9316    GO TO 9530

9318    REM ESTABLISH TAB REFERENCE (CODE B)

9320    X4=X6-X8
9322    Y4=Y6-Y8
9324    GO TO 9532

9326    REM TAB OPERATOR (CODE C)
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
9328    X6=X4+X8
9330    Y6=Y4+Y8
9332    GO TO 9530

9334    REM UNDERLINE (CODE D)

9336    PRINT @U9,21:X6+0.2667*S8*S9,Y6-0.2667*S8*C9
9338    PRINT @U9,20:X5+0.2667*S8*S9,Y5-0.2667*S8*C9
9340    GO TO 9530

9342    REM BACKSPACE (CODE E)

9344    X6=X6-X9*C9
9346    Y6=Y6-X9*S9
9348    GO TO 9530

9350    REM CARRIAGE RETURN (CODE F)

9352    X9=X8+1.5*S8*S9
9354    Y9=Y8-1.5*S8*C9
9356    X6=X9
9358    Y6=Y9
9360    X8=X9
9362    Y8=Y9
9364    GO TO 9530

9366    REM ITALICS MODE (CODE G)

9368    I9=1
9370    GO TO 9532

9372    REM NO ITALICS (CODE H)

9374    I9=0
9376    GO TO 9532
9378    P9=2
9380    M9=C8*2+1
9382    E$=SEG(N$,M9,1)
9384    F$=SEG(N$,M9+1,1)
9386    M9=ASC(E$)*16+ASC(F$)
9388    D9=C8*4+1
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
9390    E$=SEG(D$,D9,1)
9392    F$=SEG(D$,D9+1,1)
9394    G$=SEG(D$,D9+2,1)
9396    W8=D9
9398    D9=ASC(E$)*256+ASC(F$)*16+ASC(G$)
9400    D9=D9*2+1
9402    E$=SEG(D$,W8+3,1)
9404    W8=ASC(E$)
9406    FOR M9=D9 TO D9+2*N9-1 STEP 2
9408        X$=SEG(M$,M9,1)
9410        X7=ASC(X$)
9412        Y$=SEG(M$,M9+1,1)
9414        Y7=ASC(Y$)

9416        REM CHECK FOR SPECIAL CODE

9418        IF X7=15 THEN 9486

9420        REM OFFSET FOR NONCENTERED SYMBOLS

9422        X7=X7-1
9424        Y7=Y7-3

9426        REM CHECK FOR CENTERED SYMBOLS

9428        IF C8>15 THEN 9438

9430        REM OFFSET FOR CENTERED SYMBOLS

9432        Y7=Y7-6
9434        X7=X7-6

9436        REM ITALICIZE IT

9438        IF I9=0 THEN 9444
9440        X7=X7+0.3*Y7

9442        REM ROTATE IT

9444        IF A=0 THEN 9456
9446        X9=-Y7*S9+X7*C9
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
9448        Y9=Y7*C9+X7*S9
9450        X7=X9
9452        Y7=Y9

9454        REM SCALE IT

9456        X9=X7*0.06667*S8+X6
9458        Y9=Y7*0.06667*S8+Y6
9460        X7=X9
9462        Y7=Y9

9464        REM CHECK FOR A MOVE

9466        IF P9=2 THEN 9476

9468        REM DRAW IT

9470        PRINT @U9,20:X7,Y7
9472        GO TO 9480

9474        REM MOVE TO IT

9476        PRINT @U9,21:X7,Y7

9478        REM RESET DRAW CODE

9480        P9=1
9482        GO TO 9512

9484        REM SPEC CODE X7=15

9486        GO TO Y7+1 OF 9494,9510,9502
9488        GO TO 9512

9490        REM MOVE CODE
9492        REM Y7=0

9494        P9=2
9496        GO TO 9512

9498        REM Y7=2
```

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

```
9500        REM INCREASE SCALE FACTOR

9502        S8=S8*2
9504        GO TO 9512

9506        REM Y7=1
9508        REM DECREASE SCALE FACTOR

9510        S8=S8*0.5
9512    NEXT M9

9514    REM CHECK FOR CENTERED SYMBOL (RETURN TO CENTER OF SYMBOL)

9516    IF C8(16 THEN 9530

9518    REM CALCULATE WIDTH OF CHARACTER

9520    X9=W8*0.07567*S8

9522    REM CALCULATE NEXT LOCATION

9524    Y6=Y6+X9*S9
9526    X6=X6+X9*C9

9528    REM MOVE TO IT

9530    PRINT @U9,21:X6,Y6
9532 NEXT L9
9534 RETURN
```

TITLE	ABSTRACT NO:
Leroy Character Generator	51/00-8012/1

DATA FILE CONTENTS

M$ CONTAINS 512 VALUES (ASCII CODE):

```
 0 11  1  3  0 10  0 11  0  5  0  5  0 10  0  5  0 12  0 13  0  5  0 12  0 13  0
 2  0 11  0  6  0  9  0  6  0  6  0  5  0  6  0  9  0  6  1  1  0  8  0  8  0  6
 0  6  0  8  0 12  0 14  0  5  0  1  0 11  0  9  0 11  0 15  1  7  1  5  0  4  0
 6  0  6  0  8  0  5  0 10  0  2  0  8  0  2  0 13  0  5  0 14  1  0  0  4  0 14
 1  1  0  5  1  5  1  1  1  1  1  3  0  3  0  5  0  3  1  0  1 12  0  6  0 14  0
12  0  9  0  7  0  6  0 14  0  6  0  7  0  8  0  6  0  3  0  5  0  4  0 13  0  9
 1  2  0 11  0 15  0  5  0  9  0  3  0  5  0  5  0  6  0  4  0  4  0  2  0  4  0
 3  0  4  0  4  0 13  0 14  0 10  0 14  0 12  0  8  0 15  0  8  0  5  0  7  0  6
 0  2  0 14  0  8  0 11  0 14  0 13  0  6  0 12  0  5  0  8  0  3  0  5  0  5  0
 7  0  4  0 11  0  4  0 11  0  6  0  4  0  8  0 11  0  9  0 11  0  8  0  3  0  3
 0  5  0  7  0 12  0 11  0  9  0 12  0 13  1  1  1  3  0  8  0 11  0  8  0 11  0
14  0  4  0 10  0 10  1  5  1  5  0 14  0  9  1  1  1  1  0  8  0  8  0  1  0 11
 0  9  0 11  0 15  1  7  1  5  0  4  0  6  0  6  0  8  0  5  0 10  0  2  0  8  0
 2  0 13  0  5  0 14  1  0  0  4  0 14  1  1  0  5  1  5  1  1  1  1  1  3  0  3
 0  5  0  3  1  0  1 12  0 12  1  2  1  1  0  4  1  0  1  2  0  7  1  2  0  7  0
 1  1  1  0  7  1  0  0 15  0 13  0  8  1  2  0 12  0  7  0  9  0 14  1  1  1  0
 1  4  1  2  0  6  0  4  0  2  0  4  0  6  0  4  0  4  0 13  1  0  0 11  1  0  0
11  0 14  0  8  0 12  0  5  0  1  0 12  0  9  0 12  0  6  0 11  0 12  0 14  0 12
 0 12  0  9  0  9  0 12  0 14  1  4  0 12  1  1  0 11  0  4  0 11  0  9  0  6
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

D$ CONTAINS 1024 VALUES (ASCII CODE):

```
 0  0  0 14  0  0 11 14  0  1 14 14  0  2  8 14  0  3  3 14  0  3  8 14  0  3 13
14  0  4  7 14  0  4 12 14  0  5  8 14  0  6  5 14  0  6 10 14  0  7  6 14  0  8
 3 14  0  8  5 14  0  9  0 14  0  9  6  7  0  9 15 14  0 10  5 10  0 10 11 10  0
11  0 14  0 11  6 13  0 11 15 10  0 12  5 15  0 13  6 10  0 13 14 10  0 14  6 14
 0 14 12 14  0 15  2 10  0 15 10 10  1  0  6 10  1  1  4 10  1  1  9 10  1  1 10
 5  1  2  5  8  1  2 14 10  1  3  9 10  1  4  8 10  1  5 15 10  1  7  4  5  1  7
 8  7  1  7 14  7  1  8  4 10  1  8 12 10  1  9  1  5  1  9 11 10  1  9 13  5  1
10  5  8  1 10  7 10  1 11  4 10  1 11  9 10  1 12  7 10  1 13  7 10  1 13 11 10
 1 14  9 10  1 15 10 10  1 15 15 10  2  1  4 10  2  2  5  5  2  3  6  5  2  4  9
10  2  4 12 10  2  5  1 10  2  5  4 10  2  6  4 15  2  8  0 10  2  8  6 10  2  9
 4 10  2 10  0 10  2 10  9 10  2 11  0 10  2 11  6 10  2 12  4 10  2 12 10  4  2
13  1 10  2 13  9 10  2 13 15 10  2 14  2 12  2 14  7 10  2 14 11 10  2 15  8 10
 3  0  1 10  3  1  3 10  3  1 14 10  3  2 13 10  3  3  2 10  3  3 11 10  3  3 14
14  3  4  3 10  3  4  8 10  3  4 14 10  3  5  2  6  3  5  6  8  3  5  8  6  3  5
12 10  3  5 15 10  3  6  3  5  3  6  7  8  3  7  4  8  3  8  2  8  3  8 12  8  3
 9 10  8  3 10  6  8  3 10 14  8  3 11 13  8  3 12  5  4  3 12 10  4  3 13  1  8
 3 13  7  4  3 13  9 14  3 14  7  8  3 14 15  8  3 15 10  8  4  0  8  8  4  1  5
 7  4  1 11  8  4  2  7  6  4  2 12  8  4  3  4  8  4  3  7 14  4  3 12  8  4  4
 1  8  4  4  8  8  4  4 12  6  4  5  7  8  4  5 11  6  4  6  6 10  4  6 15 10  4
 7  3 10  4  7 11 10  4  8  6 10  4  8 15 14  4  9 10 10  4 10  2 10  4 10  5 10
 4 10  8 10  4 10 13 10  4 11  4 10  4 12  0 10  4 12 11 15  4 13  4 15  4 14  0
15  4 14 13 10  4 15 14 10  5  1  1 10  5  1  9 10  5  2  4 10  5  2 12 10  5  3
 7 10  5  4  5 10  5  4  9  5  5  5  3  5  5  5 13  8  5  7  2  8  5  8  7  8  5
 9  5 10  5  9 14 12  5 10 15 12  5 12  0 10  5 12  8 10  5 13  0 10  5 13  1  5
 5 13 12  8  5 14  5 10  5 15  0 10  5 15 15 10  6  1  6 10  6  2 11  5  6  2 15
 7  6  3  5  7  6  3 11 10  6  4  3 10  6  4  8  5  6  5  2 10  6  5  4  5  6  5
12  8  6  5 14 10  6  6 11 10  6  7  0 10  6  7 14 10  6  8 14 10  6  9  2 10  6
10  0 10  6 11  1 10  6 11  6 10  6 12 11 10  6 13 12  5  6 14 13  5  7  0  0 10
 7  0  3 10  7  0  8 10  7  0 11 10  7  1 11 15  7  3  7 10  7  4  3 10  7  5  5
10  7  6  6 10  7  6 10 10  7  7 10 10  7  8 12 10  7  9  3 10  7 10  5  4  7 10
12 10  7 10 13 10  7 11 14 10  7 12  5 12  7 13  5 10  7 14  4 10  7 15  1 10  7
15  9 12  8  0 11 10  8  1  7 10  8  1 14 10  8  2  7 10  8  3  5 10  8  4  6 10
 8  5  6 12  8  6 10 10  8  7 12 10  8  8  2  6  8  8  6  8  8  8  8  6  8  8 12
14  8  9  2 10  8  9  6  5  8  9 10 10  8 10  7 10  8 11  7 10  8 12  2 10  8 13
 2  7  8 13 13 10  8 14 11 10  8 15  3 10  8 15 15  4  9  0  4 10  9  0  5 10  9
 1  1 10  9  1 10 10  9  2  6 10  9  2 12 10  9  3  7 10  9  4  3 10  9  5  1 10
 9  5 13 10  9  6  9 10  9  7  2 10  9  7 11 10  9  8  7 13  9  9  5  7  9 10  9
10  9 11  5  8  9 12  6  6  9 13  1  8  9 13  5  6  9 14  0 10  9 14  9 14
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

M$ CONTAINS 5086 VALUES (ASCII CODE):

```
 2  4  2 14 12 14 12  4  2  4 15  0  6  9  8  9 15  0  7 10  7  8  6  9  8  9 15
 0  7 10  7  8 15  0  1  9  2 12  4 14  7 15 10 14 12 12 13  9 12  6 10  4  7  3
 4  4  2  6  1  9  2  6  7 15 12  6  2  6 15  0  6  9  8  9 15  0  7 10  7  8  6
 9  8  9 15  0  7 10  7  8 15  0  1  9  7 15 13  9  7  3  1  9  2  4 12 14 15  0
 2 14 12  4  1  9 13  9 15  0  7 15  7  3  7  3  2 12 12 12  7  3 15  0  6  9  8
 9 15  0  7 10  7  8 12  4  2 14 12 14  2  4 12  4  1  9  4  9  7  6  7  3  7  6
10  9 13  9 10  9  7 12  7 15  7 12  4  9  4  6  4 15  7 14 10 12 12  9 13  6  4
 6 15  0  6  9  8  9 15  0  7 10  7  8  2  4 12 14 12  4  2 14  2  4  2  4  5  7
 9  7 12  4  9  7  9 11 12 14  9 11  5 11  2 14  5 11  5  7 10 12  1 12  2  9  4
 6  7  4 10  3 10 12 15  0  6  9  8  9 15  0  7 10  7  8  7 15  7  3  1  5 13  5
10 14  4 14  1  5 15  0  6  9  8  9 15  0  7 10  7  8  3  4  7 15 11  4  1 11 13
11  3  4  2 13  2 14  3 15  4 15  5 14  5 13  4 12  3 12  2 13  7 15  7  3 15  0
 5  7  7  3  9  7  1  5  9  5 15  0  9  7  1 10  9 13  9 10  1 10 15  0  1  7  9
 7  1  9 13  9 15  0  9 11 13  9  9  7 15  2  6  3  5  2  0  2  2  7  0 10  5 10
 6  9 15  1  1 13  9 10  1  7 15  0  1  5  9  5  0  8  0  9  1 11  3 12  5 11  9
 6 11  5 13  6 14  8 14  9 13 11 11 12  9 11  5  6  3  5  1  6  0  8  1  2  9 13
15  0  1  9  9  9 15  0  1  6  9  6  5  6  5 14 15  0  1 10  9 10 15  0  1  4  9
 4  5 11  1  9  5  7 15  0  1  9 13  9  7  3  7 15 15  0  5 11  7 15  9 11  2  1
 3  0  4  1  5  4  6 11  7 14  8 15  9 14  1  3  2  6  3 12 15  0  1 11  2 12  9
12  7 12  6  6  6  4  7  3  8  4  1  9  9  9 15  0  5 11  6 12  5 13  4 12  5 11
15  0  5  5  6  6  5  7  4  6  5  5  1  9  3  9  5  3  7 15  9 15 15  0  2  3  3
 3  3  4  2  4  2  3 15  0  2  6  3  6  3 15  2 15  2  6  3 15  3 12  2 15  3 15
15  0  6 15  6 12  5 15  6 15  1  3  5 15 15  0  9 15  5  3 15  0  9 11  2 11 15
 0  1  7  8  7  5  2  5 15 15  0  8 12  6 13  4 13  2 12  2 10  3  9  7  8  8  7
 8  5  6  4  4  4  2  5  4 15  2 15  1 14  1 12  2 11  4 11  5 12  5 14  4 15  6
14  7 14  9 15  1  3 15  0  6  3  8  3  9  4  9  6  8  7  6  7  5  6  5  4  6  3
 9  3  8  3  7  4  3 13  3 14  4 15  5 15  6 14  6 13  5 12  2 10  1  8  1  6  2
 4  4  3  6  3  7  4  8  6  8  9  7  9  9  9  3 15  3 12  2 15  3 15  6 15  4 13
 3 10  3  6  4  3  6  1  1 15  3 13  4 10  4  6  3  3  1  1  1  9  9  9 15  0  7
 5  3 13 15  0  7 13  3  5  1  9  9  9 15  0  5 13  5  5  2  3  3  3  3  4  2  4
 2  3  3  4  2  4  3  3  3  2  2  0  1  9  9  9  2  3  3  3  3  4  2  4  2  3  3
 4  2  4  3  3  1  3  7 15  6  3  4  3  2  5  1  8  1 10  2 13  4 15  6 15  8 13
 9 10  9  8  8  5  6  3  2 12  5 15  5  3  2  3  8  3  1 11  1 12  2 14  4 15  6
15  8 14  9 12  8 10  7  9  3  8  2  7  1  5  1  3  9  3  2 14  4 15  6 15  8 14
 8 12  6 10  4 10  6 10  8  9  9  7  9  6  8  4  6  3  4  3  2  4  1  6  7  3  7
15  1  6  9  6  1  5  2  4  4  3  6  3  8  4  9  6  9  8  8 10  6 11  4 11  2 10
 1  9  1 15  9 15  1  8  2 10  4 11  6 11  8 10  9  8  9  6  8  4  6  3  4  3  2
 4  1  6  1 10  2 13  4 15  6 15  8 14  1 15  9 15  5  9  4  6  4  3  2 14  4 15
 6 15  8 14  8 12  6 10  4 10  6 10  8  9  9  7  9  6  8  4  6  3  4  3  2  4  1
 6  1  7  2  9  4 10  2 12  2 14  2  4  4  3  6  3  8  5  9  8  9 12  8 14  6 15
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
 4 15  2 14  1 12  1 10  2  8  4  7  6  7  8  8  9 10  2  9  3  9  3 10  2 10  2
 9  3 10  2 10  3  9 15  0  2  3  3  3  3  4  2  4  2  3  3  4  2  4  3  3  2  9
 3  9  3 10  2 10  2  9  3 10  2 10  3  9 15  0  2  3  3  3  3  4  2  4  2  3  3
 4  2  4  3  3  3  2  2  0  9 14  1  9  9  4  9 10  1 10 15  0  1  7  9  7  1 14
 9  9  1  4  2 10  1 12  2 14  4 15  6 15  8 14  9 12  8 10  6  9  5  7  5  6 15
 0  5  4  4  3  6  3  5  4 10  5  9  4  8  4  6  5  5  7  5  9  6 11  8 12  9 12
10 11 10  5 11  4 12  4 14  6 14  9 13 12 11 14  8 15  7 15  4 14  2 12  1  9  1
 7  2  4  4  2  7  1  8  1 11  2  1  3  5 15  9  3 15  0  3  7  7  7  1  3  1 15
 5 15  7 14  8 12  7 10  5  9  1  9  6  9  8  8  9  6  8  4  6  3  1  3  9 12  8
14  6 15  4 15  2 13  1 10  1  8  2  5  4  3  6  3  8  4  9  6  1  3  1 15  5 15
 8 13  9 10  9  8  8  5  5  3  1  3  9 15  1 15  1 10  6 10  1 10  1  3  9  3  9
15  1 15  1 10  6 10  1 10  1  3  9 12  8 14  6 15  4 15  2 13  1 10  1  8  2  5
 4  3  6  3  8  4  9  6  9  8  6  8  1  3  1 15  1 10  9 10  9 15  9  3  2  3  2
15  1 15  3 15 15  0  1  3  3  3  9 15  9  6  8  4  6  3  4  3  2  4  1  6  1  8
 1 15  1  3  1  7  9 15  5 11  9  3  1 15  1  3  9  3  1  3  1 15  6  3 11 15 11
 3  1  3  1 15  9  3  9 15  6  3  4  3  2  5  1  8  1 10  2 13  4 15  6 15  8 13
 9 10  9  8  8  5  6  3  1  3  1 15  5 15  8 14  9 12  9 11  8  9  5  8  1  8  6
 3  4  3  2  5  1  8  1 10  2 13  4 15  6 15  8 13  9 10  9  8  8  5  6  3 15  0
 4  6  6  5  7  3  9  2  1  3  1 15  5 15  8 14  9 12  9 11  8  9  5  8  1  8  5
 8  9  3  9 12  8 14  6 15  4 15  2 14  1 12  2 10  6  9  8  8  9  6  8  4  6  3
 4  3  2  4  1  6  1 15  9 15 15  0  5 15  5  3  9 15  9  6  8  4  6  3  4  3  2
 4  1  6  1  8  1 15  1 15  5  3  9 15  1 15  4  3  7 15 10  3 13 15  1  3  9 15
15  0  1 15  9  3  1 15  5 10  9 15 15  0  5 10  5  3  1 15  9 15  1  3  9  3  5
15  2 15  2  3  5  3  1 15  7  3  1 15  4 15  4  3  1  3  1  4  5 14  9  4 15  2
 0  1  5  1 15  1  3 15  3 12  2 15  3 15  7  6  6  4  4  3  2  4  1  6  1  9  2
11  4 12  6 11  7  9  7  6  7  3  7 12  1 15  1  3 15  0  7  9  7  6  6  4  4  3
 2  4  1  6  1  9  2 11  4 12  6 11  7  9  7  6  6  4  4  3  2  4  1  6  1  9  2
11  4 12  6 11  7  9  7  6  6  4  4  3  2  4  1  6  1  9  2 11  4 12  6 11  7  9
 7  6  7  3  7 12  7 15  7  6  6  4  4  3  2  4  1  6  1  9  2 11  4 12  6 11  7
 9  7  8  1  8  1  8  5  8 15  0  7 13  6 15  4 15  3 13  3  3  7 12  7  6  6  4
 4  3  2  4  1  6  1  9  2 11  4 12  6 11  7  9  7  3  6  1  4  0  2  1  1 15  1
 3  1  9  2 11  4 12  6 11  7  9  7  3  2 15  2 14 15  0  2 12  2  3  2 15  2 14
15  0  2 12  2  3  2  1  1  0  1 15  1  3  1  6  7 12  3  8  7  3  2 15  2  3  1
12  1  3  1  9  2 11  4 12  6 11  7  9  7  3  7  9  8 11 10 12 12 11 13  9 13  3
 1 12  1  3  1  9  2 11  4 12  6 11  7  9  7  3  7  6  6  4  4  3  2  4  1  6  1
 9  2 11  4 12  6 11  7  9  7  6  1 12  1  0 15  0  7  9  6 11  4 12  2 11  1  9
 1  6  2  4  4  3  6  4  7  6  7  9  7  9  6 11  4 12  2 11  1  9  1  6  2  4  4
 3  6  4  7  6  7  9  7 12  7  0  1 12  1  3  1  9  2 11  4 12  6 11  1  4  3  3
 5  3  7  4  7  6  6  7  2  8  1  9  1 11  3 12  5 12  7 11  3 14  3  3 15  0  1
10  5 10  1 12  1  6  2  4  4  3  6  4  7  6  7 12  7  3  1 12  4  3  7 12  1 12
```

TEKTRONIX 4051 PROGRAM

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
 4  3  7 12 10  3 13 12  1  3  7 12 15  0  1 12  7  3  1 12  4  4 15  0  7 12  4
 4  2  1  1  0  1 12  7 12  1  3  7  3  5 15  4 15  3 14  3 12  2 10  1  9  2  8
 3  6  3  4  4  3  5  3 15  2  2 11  2  1 15  1  1 15  2 15  3 14  3 12  4 10  5
 9  4  8  3  6  3  4  2  3  1  3  9 10  8  8  6  8  4 10  2 10  1  8 15  0  1  6
 9  6  5  3 10 13  0 13  5  3  9 11  1 11 15  0  9  8  1  8 15  0  9  5  1  5  9
11  1 11 15  0  9  8  1  8 15  0  9  5  1  5 15  0  1  2  9 14  9 10  8  8  6  8
 4 10  2 10  1  8 15  0  1  6  9  6 13 13 10 12  8  6  5  5  2  6  1  9  2 12  5
13  8 12 10  6 13  5  5  4  5 12 15  0  1 14  9 14 15  0  1  8  9  8  1  4  5 14
 9  4  1 14  5  4  9 14  1  5  9 13 15  0  1 13  9  5  1 15  9 15  9  3  1  3 15
 0  1  9  9  9  1 15  9 15  9  3  1  3 15  0  1  9  9  9 15  0 15  2  1  2  3 10
15  1  1 12  6 12  8 11  9  9  9  7  8  5  6  4  1  4 15  0  1  8  9  8  1 11 12
11 15  0 10 13 14  9 10  5 15  0  1  7 12  7  1 11 12 11 15  0 10 13 14  9 10  5
15  0  1  7 12  7 15  0  4  3  8 15  4 13  0  9  4  5 15  0 10  5 14  9 10 13 15
 0  2 11 12 11 15  0  2  7 12  7  2  5  8 11 15  0  2 11  8  5  6  4  4  4  2  5
 1  7  1  9  2 11  4 12  6 12  8 11  9  9  9  7  8  5  8  5  6  4  4  4  2  5  1
 7  1  9  2 11  4 12  6 12  8 11  9  9  9  7  8  5 15  0  1  8  9  8 15  0  5 12
 5  4  9 12  4 12  2 11  1  9  1  7  2  5  4  4  9  4  9 12  4 12  2 11  1  9  1
 7  2  5  4  4  9  4 15  0  3  1  8 15  1 12  6 12  8 11  9  9  9  7  8  5  6  4
 1  4  1 12  6 12  8 11  9  9  9  7  8  5  6  4  1  4 15  0  2  1  7 15  1  3  5
11 15  0  5 13  6 14  5 15  4 14  5 13 15  0  5 11  9  3 15  0  3  7  7  7  1  9
 9  9  9  6  7  6  3 14  3 15  2 15  2 14  3 14  2 15  3 15  2 14  2 13  3 11  2
14  3 14  3 15  2 15  2 14  3 15  2 15  3 14  3 13  2 11  3 14  3 15  2 15  2 14
 3 14  2 15  3 15  2 14  2 13  3 11 15  0  6 14  6 15  5 15  5 14  6 14  5 15  6
15  5 14  5 13  6 11  2 14  3 14  3 15  2 15  2 14  3 15  2 15  3 14  3 13  2 11
 .  .  .  .  .  . 15  5 15  5 14  6 15  5 15  6 14  6 13  5 11  2 15  4 14  6
12  7 10  7  6  6  4  4  3  2  4  1  6  1  8  2 10  4 11  6 10  7  8  4  9  5  8
 6  9  5 10  4  9  6  9 15  0  5 10  5  8  1  4  2  3  3  4  2  5  1  4 15  0  9
 4 10  3 11  4 10  5  9  4 15  0  5 10  6  9  7 10  6 11  5 10  1 10  2  9  3 10
 2 11  1 10 15  0  9 10 10  9 11 10 10 11  9 10 15  0  5  4  6  3  7  4  6  5  5
 4  1 15  1  4  2  2  4  1  6  1  8  2  9  4  9 15  1  1  1 12  2 14  4 15  6 15
 8 14  9 12  9  1 15  0  2  3  3  3  3  4  2  4  2  3 15  0  2  6  3  6  3 15  2
15  2  6  3 15  3 12  2 15  3 15 15  0  6 15  6 12  5 15  6 15  1  3  5 15 15  0
 9 15  5  3 15  0  9 11  2 11 15  0  1  7  8  7  5  2  5 15 15  0  8 12  6 13  4
13  2 12  2 10  3  9  7  8  8  7  8  5  6  4  4  4  2  5  4 15  2 15  1 14  1 12
 2 11  4 11  5 12  5 14  4 15  6 14  7 14  9 15  1  3 15  0  6  3  8  3  9  4  9
 6  8  7  6  7  5  6  5  4  6  3  9  3  8  3  7  4  3 13  3 14  4 15  5 15  6 14
 6 13  5 12  2 10  1  8  1  6  2  4  4  3  6  3  7  4  8  6  8  9  7  9  9  9  3
15  3 12  2 15  3 15  6 15  4 13  3 10  3  6  4  3  6  1  1 15  3 13  4 10  4  6
 3  3  1  1  1  9  9  9 15  0  7  5  3 13 15  0  7 13  3  5  1  9  9  9 15  0  5
13  5  5  2  3  3  3  3  4  2  4  2  3  3  4  2  4  3  3  3  2  2  0  1  9  9  9
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
 2  3  3  3  3  4  2  4  2  3  3  4  2  4  3  3  1  3  7 15  6  3  4  3  2  5  1
 8  1 10  2 13  4 15  6 15  8 13  9 10  9  8  8  5  6  3  2 12  5 15  5  3  2  3
 8  3  1 11  1 12  2 14  4 15  6 15  8 14  9 12  8 10  7  9  3  8  2  7  1  5  1
 3  9  3  2 14  4 15  6 15  8 14  8 12  6 10  4 10  6 10  8  9  9  7  9  6  8  4
 6  3  4  3  2  4  1  6  7  3  7 15  1  6  9  6  1  5  2  4  4  3  6  3  8  4  9
 6  9  8  8 10  6 11  4 11  2 10  1  9  1 15  9 15  1  8  2 10  4 11  6 11  8 10
 9  8  9  6  8  4  6  3  4  3  2  4  1  6  1 10  2 13  4 15  6 15  8 14  1 15  9
15  5  9  4  6  4  3  2 14  4 15  6 15  8 14  8 12  6 10  4 10  6 10  8  9  9  7
 9  6  8  4  6  3  4  3  2  4  1  6  1  7  2  9  4 10  2 12  2 14  2  4  4  3  6
 3  8  5  9  8  9 12  8 14  6 15  4 15  2 14  1 12  1 10  2  8  4  7  6  7  8  8
 9 10  2  9  3  9  3 10  2 10  2  9  3 10  2 10  3  9 15  0  2  3  3  3  3  4  2
 4  2  3  3  4  2  4  3  3  2  9  3  9  3 10  2 10  2  9  3 10  2 10  3  9 15  0
 2  3  3  3  3  4  2  4  2  3  3  4  2  4  3  3  3  2  2  0  9 14  1  9  9  4  9
10  1 10 15  0  1  7  9  7  1 14  9  9  1  4  2 10  1 12  2 14  4 15  6 15  8 14
 9 12  8 10  6  9  5  7  5  6 15  0  5  4  4  3  6  3  5  4 10  5  9  4  8  4  6
 5  5  7  5  9  6 11  8 12  9 12 10 11 10  5 11  4 12  4 14  6 14  9 13 12 11 14
 8 15  7 15  4 14  2 12  1  9  1  7  2  4  4  2  7  1  8  1 11  2  1  3  5 15  9
 3 15  0  3  7  7  7 15  0  0  3  2  3 15  0  8  3 10  3  1  3  1 15  5 15  7 14
 8 12  7 10  5  9  1  9  6  9  8  8  9  6  8  4  6  3  1  3  0  3 15  0  0 15  1
15  1  3  9 15 15  0  1 15  9  3 15  0 10 15  8 15 15  0  2 15  0 15 15  0  0  3
 2  3 15  0  8  3 10  3  9  3  1  3  5 15  9  3  9 15  1 15  1 10  6 10  1 10  1
 3  9  3  9  5 15  0  9 13  9 15  0 15 15  0  0  3  1  3  1  3  4 15  6 15  5 15
 5  3  6  3  4  3 15  0  4  6  1  7  0  9  1 11  4 12  6 12  9 11 10  9  9  7  6
 6  4  6  1  3  3  3  2  3  2 15  1 15  9 15  9 13  1  3  1 15  1 10  9 10  9 15
 9  3 15  0  0 15  2 15 15  0  8 15 10 15 15  0  0  3  2  3 15  0  8  3 10  3  2
 3  2 15  1 15  3 15 15  0  1  3  3  3 15  0  1 15  1  3  1  7  9 15  5 11  9  3
10  3  8  3 15  0  2  3  0  3 15  0  0 15  2 15 15  0  8 15 10 15  1  3  3  3  2
 3  5 15  8  3  7  3  9  3  1  3  1 15  6  3 11 15 11  3 12  3 10  3 15  0  2  3
 0  3 15  0  0 15  1 15 15  0 11 15 12 15  1  3  1 15  9  3  9 15 10 15  8 15 15
 0  2  3  0  3 15  0  0 15  1 15 15  0  8 15 10 15  6  3  4  3  2  5  1  8  1 10
 2 13  4 15  6 15  8 13  9 10  9  8  8  5  6  3  9  3  1  3  3  3  3 15  1 15  9
15  7 15  7  3  1  7  1 11  3 14  6 15  9 14 11 11 11  7  9  4  6  3  3  4  1  7
15  0  3  8  3 10  3  9  9  9  9 10  9  8  1  3  3  3  2  3  2 15  1 15  5 15  8
14  9 12  9 11  8  9  5  8  2  8  9 13  8 15  1 15  3 10  1  3  8  3  9  5  6  3
 4  3  5  3  5 15 15  0  1 13  1 15  9 15  9 13  1 14  2 15  3 15  4 14  5 11  5
 3  4  3  6  3  5  3  5 11  6 14  7 15  8 15  9 14  4  3  6  3  5  3  5 15  6 15
 4 15 15  0  0 12  1 11  1  9  2  7  4  6  6  6  8  7  9  9  9 11 10 12  2  4  2
 3  4  3  2  7  1 10  1 12  2 14  4 15  6 15  8 14  9 12  9 10  8  7  6  3  8  3
 8  4  1 12  3 14  2 13 10 13 11 14  9 12 15  0  3  8  5 10  4  9  8  9  9 10  7
 8 15  0  1  4  3  6  2  5 10  5 11  6  9  4  1  3  1 15  1 10  9 10  9 15  9  3
```

TITLE

Leroy Character Generator

ABSTRACT NO:

51/00-8012/1

```
15  0  0 15  2 15 15  0  8 15 10 15 15  0  0  3  2  3 15  0  8  3 10  3  1 13  1
15  9 15  1  3  9  3  9  5  5 15  2 15  2  3  5  3  1  3  7 15  1 15  4 15  4  3
 1  3  7  3  7 15 15  0  5 11  7 15  9 11 15  2  0  1  5  1 15  1  3 15  3 12  2
15  3 15  9 11  8  7  7  4  5  3  3  3  1  5  1  8  2 11  4 12  5 12  7 10  8  3
 9  3  1  0  3 12  4 14  6 15  8 14  9 12  8 10  6  9  3  9  6  9  8  8  9  6  9
 4  7  3  5  3  2  4  2 11  3 12  4 12  5 10  5  2  .  0  7  0  8  1 15  0  1  0
 9 12  6 11  3 10  2  9  1  6  2  4  4  3  7  .  8  5  9  8  8 10  4 12  3 13  3
14  4 15  6 15  8 14  6 12  4 12  2 11  1  9  1  8  6  8  1  8  1  6  2  4  4  3
 6  3  6 11  3 10  2  9  1  6  2  4  4  3  7  4  8  5  9  8  8 10  6 11 15  0  6
14  4  0  1  9  2 11  4 12  6 11  7  7  6  6  3  0  9 12  1 12  2 12  3 11  3  9
 2  3  3  9  4 11  6 12  8 12  9 10  9  6  8  0  2 12  1  6  1  4  2  3  3  3 15
 0  1 12  2 12  3 10  3  7  2  3  3  7  6 10  9 12  6 10  7  5  8  3  9  3  1  3
 5  9 15  0  1 15  3 13  5  9  7  5  8  3  9  4  1  0  3 12  2  6  2  4  4  3  6
 4  7  6  8 12  7  6  7  4  8  3  9  4  1 12  3 12  3  3  6  5  8  8  9 12  6 12
 8 11  9  9  9  7  7  4  4  3  2  4  1  6  1  8  3 11  6 12  1  3  2  6  3 12 15
 0  1 11  2 12  9 12  7 12  6  6  6  4  7  3  8  4  1  9  9  9  8  6  6  4  4  3
 2  4  1  6  1  9  2 12  4 14  6 15  8 14  9 12  9  9  1  0  2  6  4 11  6 12  8
11  9  9  9  7  8  4  6  3  4  3  3  4  2  6  6 12  8 11  9  9  9  7  7  4  4  3
 2  4  1  6  1  8  3 11  6 12 10 12  1 11  2 12  9 12  5 12  4  6  4  4  5  3  6
 3  7  4  1 11  2  9  1  6  2  4  4  3  6  3  8  5  8  9  9 11  1 11  2  9  1  6
 2  4  4  3  6  3  8  5  8  9  9 11 15  0  6 14  4  0  3 12  1  8  1  6  2  4  4
 3  6  4  7 10  6  4  7  3  9  3 11  4 12  7 12 10 11 12  4 15  3 15  4 14  5 14
 6 15  3 14  2 12  2 10  4  9  6  9  4  9  2  7  1  5  1  4  2  3  5  2  6  1  5
 0  4  0  2  1  1 12  2 12  3 11  3  9  2  3  3  9  4 11  6 12  8 12  9 10  9  6
 8  0  3 15  3 14  4 13  6 13  7 14  6 14  4 13  3 12  2 10  1  7  1  5  2  4  4
 3  5  2  5  1  4  0  2  1  5 15  4 15  3 14  3 12  2 10  1  9  2  8  3  6  3  4
 4  3  5  3 15  2  2 11  2  1 15  1  1 15  2 15  3 14  3 12  4 10  5  9  4  8  3
 6  3  4  2  3  1  3  9 10  8  8  6  8  4 10  2 10  1  8 15  0  1  6  9  6  7 15
 7  3 15  0  5  7  7  3  9  7
```

TEKTRONIX

Type 19 F-30

4051

APPLICATIONS LIBRARY PROGRAM

TITLE		ABSTRACT NUMBER
SPIDER'S WEB CHARACTER GENERATOR		51/00-8036/0
ORIGINAL DATE Auguts 1979	REVISION DATE	MEMORY REQUIREMENT 16K
AUTHOR Gene Lynch	Tektronix, Inc. Wilsonville, OR	PERIPHERALS Optional - 4662 Plotter

ABSTRACT

Statements: 402

Files: 1 ASCII Program

The subroutine draws characters for titles and illustrations. The curves of the characters are formed as an envelope of tangent line segments. Only standard upper case letters are supported. An example program is included to demonstrate program operation.

Program operation:

Store the characters to be drawn in P$

Store the width of a character in KØ

Store the device address in K8 (32 = screen)

Store the aspect ratio for a character in V3 (height/width)

Set a square window and viewport

Set the starting coordinates for the string to be drawn

Call the routine

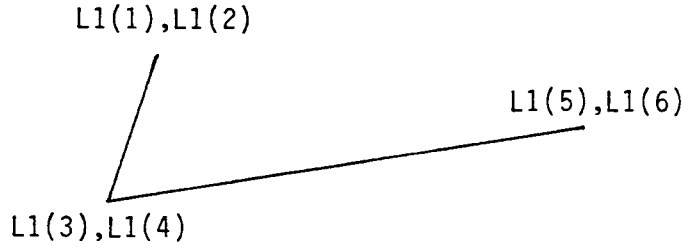
The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

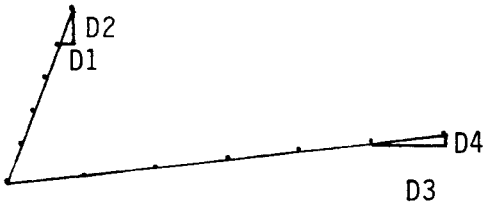
METHOD

The character set is based on the fact that when the sides of an angle are divided into an equal number of points and these points are connected in inverse order, the resulting line segments are tangent to a curve (a parabola) and form an envelope of the curve.

Step one: Three points determine an angle

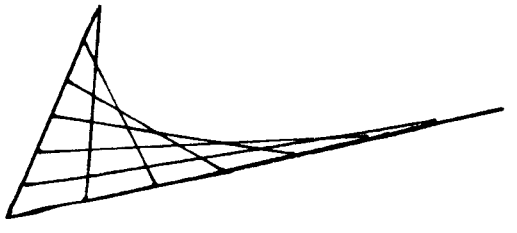


Step two: Divide the line segments into "u" equal parts



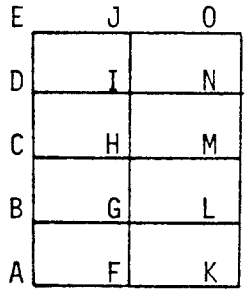
D1 = (L1(3)-L1(1))/u
D2 = (L1(4)-L1(2))/u
D3 = (L1(5)-L1(3))/u
D4 = (L1(6)-L1(4))/u

Step three: Connect the points in inverse order



TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

The characters are created from combining several of these basic elements. The present characters are encoded on a 3x5 grid. The points on the grid are identified by characters as shown.



The code for the letter "A" is AJK

The code for the letter "T" is EJFOJF

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

PROGRAM LOADING

Press AUTOLOAD if the program is located on the first file on the tape otherwise:

```
FIND #  (number of file)
OLD
RUN
```

The program will automatically generate a sample to the screen.

PROGRAM EXECUTION

The program is a subroutine. Lines prior to 1000 set the required variables:

- P$ - the characters to be drawn
- KØ - the width of a character
- K8 - the output device address (32 = screen)
- V3 - the aspect ratio for a character
- X, Y - the lower left had corner coordinates of the character
- GOSUB 4000 - calls the routine

TITLE

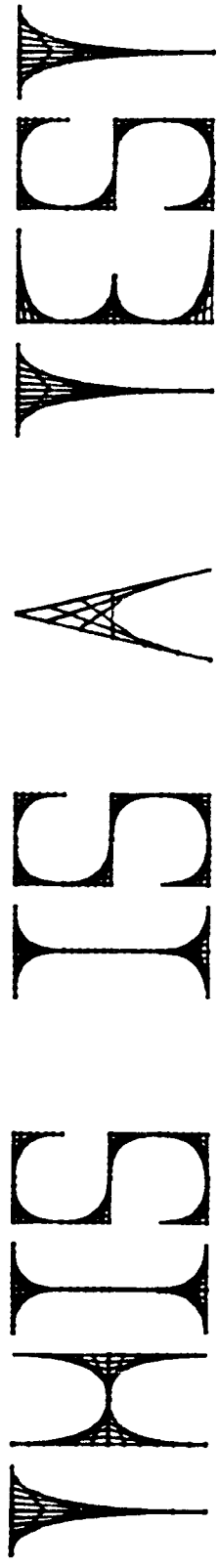
SPIDER'S WEB CHARACTER GENERATOR

ABSTRACT NO:

51/00-8036/0

THIS IS A TEST OF
THE 4050 SPIDER'S
WEB CHARACTER
GENERATOR
AUGUST 14, 1979

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0



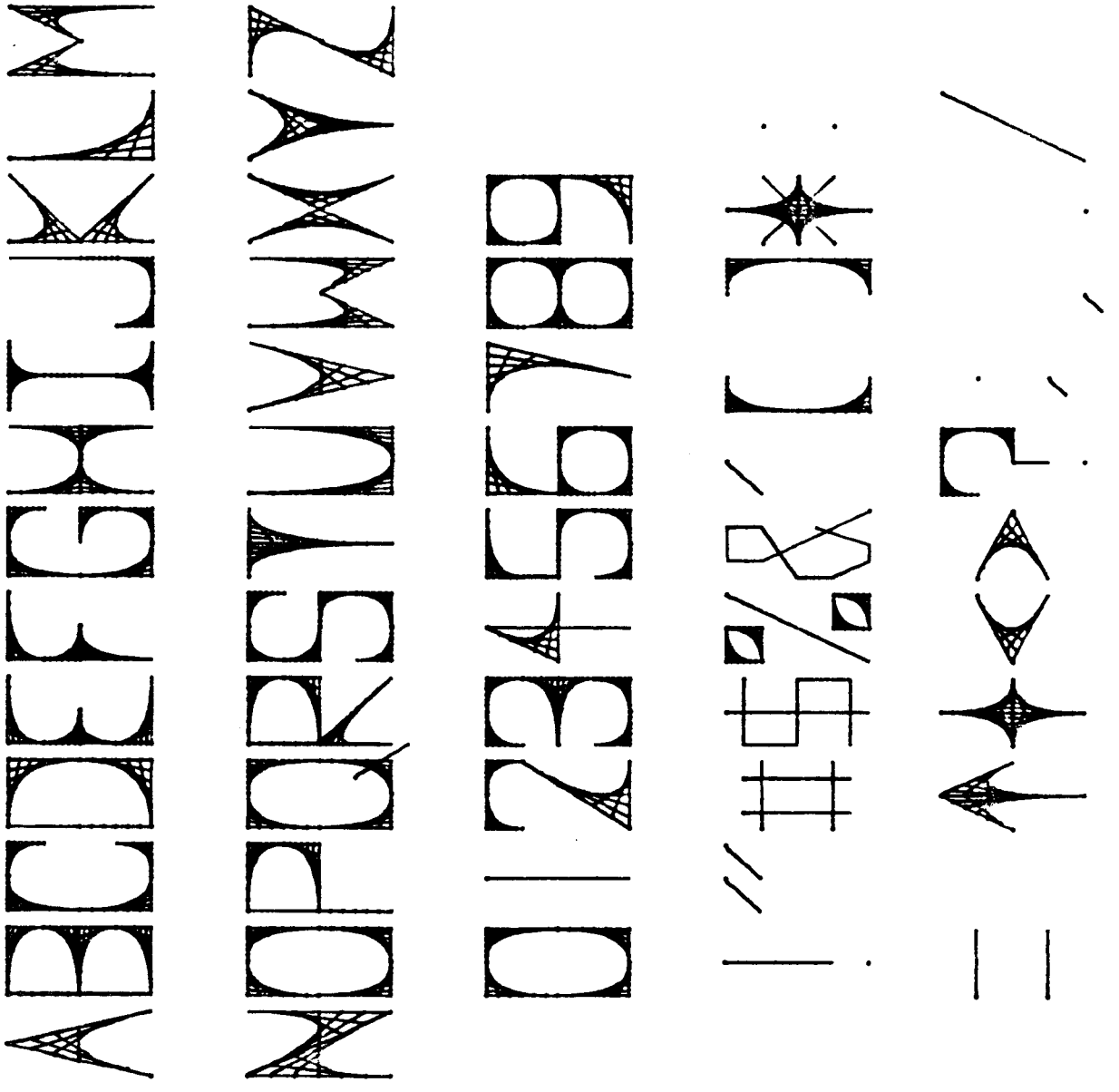

```
100 PAGE
110 WINDOW 0,130,0,130
120 VIEWPORT 0,130,0,130
130 K0=15
140 K8=1
150 U3=2
160 X=0
170 Y=60
180 P$="THIS IS A TEST"
190 GOSUB 4000
200 END
```

TITLE

SPIDER'S WEB CHARACTER GENERATOR

ABSTRACT NO:

51/00-8036/0



TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

DATA STORAGE

Variable	Used to Store	Type
A$	Segment of web code	String
C$	Segment of character string	String
C	ASCII code for a character	Simple
D1	X step difference	Simple
D2	Y step difference	Simple
D3	X step difference	Simple
D4	Y step difference	Simple
H$	Web codes for the characters	String
H	Number of characters in a character code	Simple
I1	Index in a loop	Simple
I2	Index in a loop	Simple
I8	Index in a loop	Simple
KØ	Character size	Simple
K1	1/2 character size	Simple
K2	1/4 character size	Simple
K4	Missing character or space flag	Simple
K8	Device primary address	Simple
L1	Coordinate points of a segment	Array (6)
N1	Number of characters to be drawn	Simple
P$	Character string to be drawn	String
U	Number of line segments in an element	Simple
V	Off set code for a character	Simple
V2	Aspect ratio factor	Simple
V3	Aspect ratio (height/width)	Simple
X	X coordinate (lower left corner)	Simple
XØ	X coordinate of a point in the character	Simple
Y	Y coordinate (lower left corner)	Simple
YØ	Y coordinate of a point in the character	Simple

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
100 REM 51/00-8036/0  SPIDER'S WEB CHARACTER GENERATOR
110 REM   INITIALIZATION
120 REM  K0 = CHARACTER WIDTH IN GDU'S
140 REM  K8  = DEVICE NUMBER (32 IS THE SCREEN)
150 REM  V3 = CHARACTER ASPECT RATIO (HEIGHT/WIDTH)
160 REM

170 WINDOW 0,130,0,130
180 VIEWPORT 0,130,0,130
190 PAGE
200 K0=12
230 K8=32
250 V3=2

270 REM
280 REM
500 REM   MAIN PROGRAM
510 REM
520 REM   (X,Y) ARE THE COORDINATES OF THE LOWER LEFT HAND CORNER
530 REM   OF THE CHARACTER.
540 REM  P$ = THE CHARACTER STRING TO BE DRAWN
550 REM

560 X=0
570 Y=80
580 P$="THIS IS A TEST OF"
590 GOSUB 4010
600 X=0
610 Y=60
620 P$="THE 4050 SPIDER'S"
630 GOSUB 4010
640 X=0
650 Y=40
660 P$="WEB CHARACTER _"
670 GOSUB 4010
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
680 X=0
690 Y=20
700 P$="GENERATOR"
710 GOSUB 4010
720 X=0
730 Y=2
740 P$="AUGUST 14, 1979"
750 GOSUB 4010
760 X=0
770 Y=30
780 END

790 REM
800 REM
1000 REM
1010 REM DECODING SUBROUTINE TO ESTABLISH THE COORDINATES OF A POINT
1020 REM ON THE WEB.
1030 REM

1040 C=ASC(A$)
1050 K4=0
1060 C=C-65
1070 IF C>4 THEN 1110
1080 X0=X
1090 Y0=Y+C*K2*V2
1100 RETURN
1110 IF C>9 THEN 1150
1120 X0=X+K2
1130 Y0=Y+(C-5)*K2*V2
1140 RETURN
1150 IF C>14 THEN 1190
1160 X0=X+K1
1170 Y0=Y+(C-10)*K2*V2
1180 RETURN
1190 X0=X+K1
1200 K4=1
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
1210 YO=Y
1220 RETURN

1230 REM
1240 REM
2000 REM DECODE SETS OF THREE POINTS TO FORM WEBS
2010 REM

2020 K1=K0/2
2030 K2=K1/2
2040 DELETE L1
2050 DIM L1(6)
2060 H=LEN(H$)
2070 FOR I1=1 TO H/3
2080     FOR K=1 TO 3
2090         A$=SEG(H$,1,1)
2100         H$=REP("",1,1)
2110         GOSUB 1040
2120         IF K4=1 THEN 2170
2130         L1(2*K-1)=X0
2140         L1(2*K)=Y0
2150     NEXT K
2160     GOSUB 2210
2170     MOVE @K8:X0,Y0
2180     MOVE @K8:X0,Y0
2190 NEXT I1
2200 RETURN

2205 REM  U = NUMBER OF LINES PER ENVELOPE CURVE (WEB)

2210 U=6

2220 REM
2230 REM
3000 REM  ROUTINE TO DRAW THE WEB
3010 REM
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
3020 D1=(L1(3)-L1(1))/U
3030 D2=(L1(4)-L1(2))/U
3050 D3=(L1(5)-L1(3))/U
3060 D4=(L1(6)-L1(4))/U
3070 FOR I2=0 TO U
3080     PRINT @K8,21:L1(1)+I2*D1,L1(2)+I2*D2
3090     PRINT @K8,20:L1(3)+I2*D3,L1(4)+I2*D4
3100 NEXT I2
3110 RETURN

4000 REM SELECT CHARACTER TO BE DRAWN CALL FOR ENCODING AND DRAWING

4010 N1=LEN(P$)
4012 K1=K0/2
4014 K2=K1/2
4016 SET NOCASE
4018 V2=V3/2
4020 FOR I8=1 TO N1
4030     C$=SEG(P$,1,1)
4040     P$=REP("",1,1)
4050     GOSUB 5010
4060     GOSUB 2020
4070     X=X+2.5*K2
4080     MOVE @K8:X,Y
4090 NEXT I8
4100 RETURN

4110 REM
4120 REM
5000 REM SELECT AND ENCODE WEB PATTERN FOR A GIVEN CHARACTER

5010 V=ASC(C$)-32
5020 IF V<1 THEN 5240
5030 IF V>10 THEN 5060
5040 GOSUB V OF 6840,6910,6980,7090,7200,7250,7380,7430,7460,7490
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
5050 RETURN
5060 IF V>20 THEN 5090
5070 GOSUB V-10 OF 7560,7590,7640,7690,7740,7790,7820,7870,7900,7930
5080 RETURN
5090 IF V>30 THEN 5120
5100 GOSUB V-20 OF 7980,8010,8040,8070,8100,8130,8200,8270,8300,8370
5110 RETURN
5120 IF V>32 THEN 5150
5130 GOSUB V-30 OF 8400,6810
5140 RETURN
5150 IF V>42 THEN 5180
5160 GOSUB V-32 OF 6010,6040,6070,6100,6130,6160,6190,6220,6250,6280
5170 RETURN
5180 IF V>52 THEN 5210
5190 GOSUB V-42 OF 6310,6340,6370,6400,6430,6460,6490,6540,6570,6600
5200 RETURN
5210 IF V>62 OR V<1 THEN 5240
5220 GOSUB V-52 OF 6630,6660,6690,6720,6750,6780,7430,6810,7460,8490
5230 RETURN
5240 GOSUB 6810
5250 RETURN

5260 REM
5270 REM
6000 REM WEB CODES FOR THE CHARACTERS

6010 H$="AJK"
6020 RETURN

6030 REM B

6040 H$="ŁONNMCCMLLKAŁCA"
6050 RETURN

6060 REM C
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
6070 H$="NOJJECCAFFKL"
6080 RETURN

6090 REM D

6100 H$="EOMMKAACE"
6110 RETURN

6120 REM E

6130 H$="OEDDCHHCBBAK"
6140 RETURN

6150 REM F

6160 H$="OEDDCHHCA"
6170 RETURN

6180 REM G

6190 H$="NOJJECCAFFKLLMH"
6200 RETURN

6210 REM H

6220 H$="ECHHMOACHHMK"
6230 RETURN

6240 REM I

6250 H$="EJIOJIIHGGFAGFK"
6260 RETURN

6270 REM J

6280 H$="BAFFKLLMO"
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
6290 RETURN

6300 REM K

6310 H$="ECOACK"
6320 RETURN

6330 REM L

6340 H$="EAK"
6350 RETURN

6360 REM M

6370 H$="AEHHOK"
6380 RETURN

6390 REM N

6400 H$="AEKEKO"
6410 RETURN

6420 REM O

6430 H$="CEJJOMNKFFAC"
6440 RETURN

6450 REM P

6460 H$="ACEEONNMC"
6470 RETURN

6480 REM Q

6490 MOVE @K8:X+K2*1.5,Y+K2*V2
6500 DRAW @K8:X+2.5*K2,Y-K2/2*V2
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
6510 H$="CEJJOMMKFFAC"
6520 RETURN

6530 REM R

6540 H$="EONNMCECABCK"
6550 RETURN

6560 REM S

6570 H$="NOJJEDDCHHMLLKFFAB"
6580 RETURN

6590 REM T

6600 H$="EJFFJO"
6610 RETURN

6620 REM U

6630 H$="EAFFKO"
6640 RETURN

6650 REM V

6660 H$="EFO"
6670 RETURN

6680 REM W

6690 H$="EAHHKO"
6700 RETURN

6710 REM X

6720 H$="EHAOHK"
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
6730 RETURN

6740 REM Y

6750 H$="ÆHOÆHFOHF"
6760 RETURN

6770 REM Z

6780 H$="ÆOHHAK"
6790 RETURN

6800 REM SPACÆ

6810 H$="S"
6820 RETURN

6830 REM !

6840 MOVÆ @K8:X+K2,Y+4*K2*V2
6850 DRAW @K8:X+K2,Y+K2*V2
6860 MOVÆ @K8:X+K2,Y
6870 DRAW @K8:X+K2,Y
6880 H$="S"
6890 RETURN

6900 REM "

6910 MOVÆ @K8:X+K2,Y+4*K2*V2
6920 DRAW @K8:X,3*K2*V2+Y
6930 MOVÆ @K8:X+K1,Y+4*K2*V2
6940 DRAW @K8:X+K2,Y+3*K2*V2
6950 H$="S"
6960 RETURN

6970 REM #
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
6980 MOVE @K8:X+K2/2,Y+3.5*K2*V2
6990 DRAW @K8:X+K2/2,Y+0.5*K2*V2
7000 MOVE @K8:X+1.5*K2,Y+3.5*K2*V2
7010 DRAW @K8:X+1.5*K2,Y+0.5*K2*V2
7020 MOVE @K8:X,Y+3*K2*V2
7030 DRAW @K8:X++K1,Y+K2*V2*3
7040 MOVE @K8:X,Y+K2*V2
7050 DRAW @K8:X+K1,Y+K2*V2
7060 H$="S"
7070 RETURN

7080 REM $

7090 MOVE @K8:X+K1,Y+3.5*K2*V2
7100 DRAW @K8:X,3.5*K2*V2+Y
7110 DRAW @K8:X,Y+2*K2*V2
7120 DRAW @K8:X+K1,Y+2*K2*V2
7130 DRAW @K8:X+K1,Y+0.5*K2*V2
7140 DRAW @K8:X,Y+0.5*K2*V2
7150 MOVE @K8:X+K2,Y+4*K2*V2
7160 DRAW @K8:X+K2,Y
7170 H$="S"
7180 RETURN

7190 REM %

7200 MOVE @K8:X+K1,Y+4*K2*V2
7210 DRAW @K8:X,Y
7220 H$="DEJJIDFGLLKF"
7230 RETURN

7240 REM &

7250 MOVE @K8:X+K1,Y
7260 DRAW @K8:X+K2/2,Y+3*K2*V2
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
7270 DRAW @K8:X+K2/2,Y+4*K2*V2
7280 DRAW @K8:X+K2*1.5,Y+4*K2*V2
7290 DRAW @K8:X+K2*1.5,Y+3*K2*V2
7300 DRAW @K8:X,Y+2*K2*V2
7310 DRAW @K8:X,Y+K2*V2
7320 DRAW @K8:X+K2/2,Y
7330 DRAW @K8:X+K2,Y
7340 DRAW @K8:X+K2*1.5,Y+1.5*K2*V2
7350 H$="S"
7360 RETURN

7370 REM '

7380 MOVE @K8:X+K2,Y+4*K2*V2
7390 DRAW @K8:X,Y+3*K2*V2
7400 H$="S"
7410 RETURN

7420 REM (

7430 H$="JECCAF"
7440 RETURN

7450 REM )

7460 H$="JOMMKF"
7470 RETURN

7480 REM *

7490 MOVE @K8:X,Y+3*K2*V2
7500 DRAW @K8:X+K1,Y+K2*V2
7510 MOVE @K8:X,Y+K2*V2
7520 DRAW @K8:X+K1,Y+K2*V2*3
7530 H$="MHJJHCCHFFHM"
7540 RETURN
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
7550 REM +

7560 H$="MHJJHCCHFFHM"
7570 RETURN

7580 REM ,

7590 MOVE @K8:X+K2,Y
7600 DRAW @K8:X+K2/2,Y-K2*V2*0.5
7610 H$="S"
7620 RETURN

7630 REM -

7640 MOVE X,Y+2*K2*V2
7650 DRAW X+K1,Y+2*K2*V2
7660 H$="S"
7670 RETURN

7680 REM .

7690 MOVE @K8:X+K2,Y
7700 DRAW @K8:X+K2,Y
7710 H$="S"
7720 RETURN

7730 REM /

7740 MOVE @K8:X+K1,Y+4*K2*V2
7750 DRAW @K8:X,Y
7760 H$="S"
7770 RETURN

7780 REM 0
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
7790 H$="CEJJOMMKFFAC"
7800 RETURN

7810 REM 1

7820 MOVE @K8:X+K2,Y+4*K2*V2
7830 DRAW @K8:X+K2,Y
7840 H$="S"
7850 RETURN

7860 REM 2

7870 H$="DEJJONNAK"
7880 RETURN

7890 REM 3

7900 H$="DEJJONNMCCMLLKFFAB"
7910 RETURN

7920 REM 4

7930 MOVE @K8:X+K2,Y+4*K2*V2
7940 DRAW @K8:X+K2,Y
7950 H$="JCM"
7960 RETURN

7970 REM 5

7980 H$="OEDEDCCHMHMLLKFFAB"
7990 RETURN

8000 REM 6

8010 H$="OECBCHHMLLKFFAB"
8020 RETURN
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
8030 REM 7

8040 H$="EOF"
8050 RETURN

8060 REM 8

8070 H$="DEJJONNMHHCDBCHHMLLKFFAB"
8080 RETURN

8090 REM 9

8100 H$="DEJJONNMHHCDMKA"
8110 RETURN

8120 REM :

8130 MOVE @K8:X+K2,Y+3*K2*V2
8140 DRAW @K8:X+K2,Y+3*K2*V2
8150 MOVE @K8:X+K2,Y+K2*V2
8160 DRAW @K8:X+K2,Y+K2*V2
8170 H$="S"
8180 RETURN

8190 REM ;

8200 MOVE @K8:X+K2,Y+K2*3*V2
8210 DRAW @K8:X+K2,Y+3*K2*V2
8220 MOVE @K8:X+K2,Y+K2*V2
8230 DRAW @K8:X+K2/2,Y+K2*V2/2
8240 H$="S"
8250 RETURN

8260 REM <
```

TITLE	ABSTRACT NO:
SPIDER'S WEB CHARACTER GENERATOR	51/00-8036/0

```
8270 H$="NCL"
8280 RETURN

8290 REM =

8300 MOVE @K8:X,Y+3*K2*V2
8310 DRAW @K8:X+K1,Y+3*K2*V2
8320 MOVE @K8:X,Y+K2*V2
8330 DRAW @K8:X+K1,Y+K2*V2
8340 H$="S"
8350 RETURN

8360 REM >

8370 H$="DMB"
8380 RETURN

8390 REM ?

8400 MOVE @K8:X+K2,Y+2*K2*V2
8410 DRAW @K8:X+K2,Y+K2*V2
8420 MOVE @K8:X+K2,Y
8430 DRAW @K8:X+K2,Y
8440 H$="DEJJONNMH"
8450 RETURN

8460 REM
8470 REM
8480 REM ↑

8490 H$="CJFFJM"
8500 RETURN
```

TEKTRONIX

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0
ORIGINAL DATE: August 1978 / REVISION DATE:	MEMORY REQUIREMENT: 16K
AUTHOR: Roger Chan, USV Pharmaceutical Corp. Tuckahoe, NY	PERIPHERALS: 4631 Hard Copy Unit

ABSTRACT

Statements: 129

Files: 1 ASCII Program

The program reads any size ASCII tape file program or data and lists it, line by line, on the screen. When the page is full, or last line is encountered, a hard copy on the 4631 is automatically produced. All control characters are processed so they will not execute when read and printed to the screen. The file numbers are noted on the first page of the listing.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

PROGRAM LOADING

Press AUTOLOAD if the program is located on the first file on the tape, otherwise:

```
FIND # ( file number )
OLD
RUN
```

Respond to the questions as required.

NOTE: If the program is interrupted in the middle, the screen may be set back to normal by the following statement:

```
PRINT @32,26:0
```

TITLE	ABSTRACT NO:
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

EXAMPLE

```
PLEASE PUT IN TAPE TO BE PRINTED & PRESS RETURN



STARTING  FROM FILE =1
          END FILE =1
INPUT TAPE NAME:MASTER TAPE
```

```
TAPE NAME IS MASTER TAPE                FILE NO. IS 1

DISPLAY CHARACTER CR=M


69 REM **** 51/00-8037/0 ***** AUTOMATIC HARDCOPY FILE LISTINGS ********
70 REM ************************************************************
80 REM **** THIS IS A PROGRAM TO  OUTPUT BEAUTIFUL HARDCOPY FROM *******
83 REM **** TAPE USING 4051 MACHINE AND A 4631 HARD COPIER **************
85 REM **** THE WHOLE PROCESS IS AUTOMATIC  WITH MINIMUM ATTENTION *****
87 REM **** FROM OPERATOR  AND IT IS FAST ******************************
90 REM **** WRITTEN BY ROGER CHAN ON AUG. 16, 1978; 3 A.M. **************
95 REM ************************************************************
96 REM
97 REM
98 REM
100 INIT
101 PAGE
105 REM **** A$ IS UNDERSCORE CHARACTER ****
110 A$=CHR(95)
120 B$=" "
130 C$=" "
140 K=1
150 I=0
151 PRINT "PLEASE PUT IN TAPE TO BE PRINTED & PRESS RETURN"
152 INPUT B$
153 PAGE
155 REM **** INPUT BEGINNING AND ENDING FILE NO. ****
160 PRINT "STARTING  FROM FILE =";
170 INPUT N
180 PRINT "          END FILE =";
190 INPUT N1
200 F=0
210 DIM L$(99)
215 REM **** SET PAGE FULL CONDITION TO AUTO PAGE AND COPY ****
```

TITLE

AUTOMATIC HARDCOPY FILE LISTINGS

ABSTRACT NO:

51/00-8037/0

TITLE	ABSTRACT NO:
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

DATA STORAGE

Variable	Used to Store	Type
A$	Underscore character	String
B$	Control character	String
C$	Replacing characters	String
F	Logic flag	Simple
G$	Working storage	String
I	Positional counter	Simple
J	New statement's length	Simple
K	Search counter	Simple
L$	Input storage	String (99)
N	Starting file number	Simple
N1	Ending file number	Simple
P	Branching control	Simple
T$	Tape name	String

TITLE	ABSTRACT NO:
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

```
69 REM **** 51/00-8037/0 ***** AUTOMATIC HARDCOPY FILE LISTINGS ********
70 REM ************************************************************
80 REM **** THIS IS A PROGRAM TO  OUTPUT BEAUTIFUL HARDCOPY FROM *******
83 REM **** TAPE USING 4051 MACHINE AND A 4631 HARD COPIER *************
85 REM **** THE WHOLE PROCESS IS AUTOMATIC  WITH MINIMUM ATTENTION *****
87 REM **** FROM OPERATOR  AND IT IS FAST ******************************
90 REM **** WRITTEN BY ROGER CHAN ON AUG. 16, 1978; 3 A.M. *************
95 REM ************************************************************
96 REM
97 REM
98 REM

100 INIT
101 PAGE

105 REM **** A$ IS UNDERSCORE CHARACTER ****

110 A$=CHR(95)
120 B$=" "
130 C$=" "
140 K=1
150 I=0
151 PRINT "PLEASE PUT IN TAPE TO BE PRINTED & PRESS RETURN"
152 INPUT B$
153 PAGE

155 REM **** INPUT BEGINNING AND ENDING FILE NO. ****

160 PRINT "STARTING  FROM FILE =";
170 INPUT N
180 PRINT "          END FILE =";
190 INPUT N1
200 F=0
210 DIM L$(99)
```

TITLE	ABSTRACT NO.
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

```
215 REM **** SET PAGE FULL CONDITION TO AUTO PAGE AND COPY ****

220 PRINT @32,26:3
230 PRINT "INPUT TAPE NAME:";

235 REM **** T IS THE TAPE NUMBER WHICH IS A CONSTANT ****

240 INPUT T$

245 REM **** FILE COUNTER ****

250 FIND N

265 REM **** PRINT LAST PAGE OF THE FILE ****

270 ON EOF (0) THEN 840
280 PAGE

285 REM **** BEGINING OF OUTPUT ****
286 REM **** FIRST LINE IS TAPE NO. & FILE NO. ****
287 REM **** THIRD LINE IS DISPLAY CHARACTER FOR CR ****

290 PRINT "TAPE NAME IS ";T$,"FILE NO. IS ";N
300 PRINT "_DISPLAY CHARACTER CR=MH";A$;"__"

305 REM **** READ A LINE FROM TAPE ****

310 INPUT @33:L$

315 REM **** SEARCH FOR PRINT STATEMENTS FOR SPECIAL HANDLING ****

320 I=POS(L$," ",1)
330 G$=SEG(L$,I+1,3)
340 IF G$="PRI" THEN 370

341 REM **** SEARCH FOR ASSIGNMENT STRING FOR SPECIAL HANDLING***
```

TITLE	ABSTRACT NO:
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

```
345 G$=SEG(L$,I+2,1)
346 IF G$="$" THEN 370
350 PRINT L$
360 GO TO 310

365 REM **** ROUTINE TO HANDLE PRINT STATEMENT ****
367 REM **** PRINT STATEMENT WILL BE SEARCH CHARACTER BY CHARACTER ****
368 REM **** FOR SPECIAL CHARACTERS ****

370 J=LEN(L$)
380 K=1
385 F=0
390 IF K=J THEN 350
400 GOSUB 500
410 IF F=0 THEN 430
420 K=K+2
425 GO TO 450
430 K=K+1
450 F=0
460 GO TO 390

495 REM **** SUBROUTINE  TO TEST FOR SPECIAL CHARACTERS ****

500 T$=SEG(L$,K,1)
510 P=ASC(T$)
520 IF P=>7 AND P<=13 THEN 560

525 REM **** TEST FOR CRTL ↑ AND CRTL RUBOUT ****

530 IF P=30 OR P=31 THEN 770
540 F=0
550 RETURN

555 REM **** BRANCHING FOR ALL SPECIAL CHARACTERS EXCEPT  RS & CR/LF ***
```

TITLE	ABSTRACT NO:
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

```
560 P=P-6
570 GOSUB P OF 630,650,670,690,710,730,750
580 P=0

585 REM **** COMBINE WHATEVER CHARACTER WITH THE UNDERSCORE ****

590 C$=B$&A$

595 REM **** REPLACE INPUT LINE WITH NEWLY COMBINED STRING ****

600 L$=REP(C$,K,1)
610 F=1
615 J=J+2
620 RETURN

625 REM ****  CTRL G ****

630 B$="GH"
640 RETURN

645 REM ****  CTRL H ****

650 B$="HH"
660 RETURN

665 REM ****  CTRL I ****

670 B$="IH"
680 RETURN

685 REM ****  CTRL J ****

690 B$="JH"
700 RETURN

705 REM ****  CTRL K ****
```

TITLE	ABSTRACT NO
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

```
710 B$="KH"
720 RETURN

725 REM ****  CTRL L ****

730 B$="LH"
740 RETURN

745 REM ****  CTRL M ****

750 B$="MH"
760 RETURN
770 IF P=30 THEN 820

775 REM ****  CTRL RUBOUT ****

780 B$=" H"
790 C$=B$&A$
800 L$=REP(C$,K,1)
810 GO TO 610

815 REM ****  CTRL ↑ ****

820 B$="↑H"
830 GO TO 590
840 N=N+1
845 FIND N
850 PRINT @32,26:3
860 COPY

865 REM **** TEST  TO SEE IF IT IS LAST FILE REQUESTED ****

870 IF N>N1 THEN 900
880 GO TO 250
900 PRINT @32,26:0
```

TITLE	ABSTRACT NO:
AUTOMATIC HARDCOPY FILE LISTINGS	51/00-8037/0

```
910 END

1000 REM **** YOU JUST HAVE A WHOLE SET OF PRINTOUT FROM TAPE ****
1001 REM **** ALMOST FOR NO EFFORT AT ALL ****
```

TEKTRONIX

TAPE 19 F-32

4051

APPLICATIONS LIBRARY PROGRAM

TITLE		ABSTRACT NUMBER
TEXT EDITOR		51/00-8038/0
ORIGINAL DATE February 1980	REVISION DATE	MEMORY REQUIREMENT 32K
AUTHOR Dennis Ward	Tektronix, Inc. Beaverton, OR	PERIPHERALS None

ABSTRACT

Statements: 262

Files: 1 ASCII Program
Requires pre-MARked file

The program edits text or program files without using syntax-dependent, complicated command forms. The commands are entered by number:

1 - Enter text from keyboard or get from tape file

2 - List lines of text (all or part)

3 - Delete lines of text (line #'s)

4 - Insert lines (insert before line #)

5 - Swap lines (input 2 line numbers)

6 - Search and replace characters (start at beginning of line #)

7 - Renumber lines (start at line # and increment)

8 - Save on tape (pre-MARked file #)

9 - Quit

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO.
TEXT EDITOR	51/00-8038/0

METHOD

The program was written for people who do not wish to learn another set of edit commands. All commands are entered by number.

The program is large, using about 20K bytes. The text/program file created by the TEXT EDITOR is therefore limited to about 8,000 to 9,000 characters in a machine with 32K memory.

The program is powerful when combined with the editing keys of the machine.

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

PROGRAM LOADING

Press AUTOLOAD is the program is located on the first file on the tape, otherwise:

FIND # (file number)

OLD

RUN

Respond to the questions as required.

PROGRAM OPERATION

Text may be entered from the keyboard or from a file.

Data files must be pre-MARKed before saving text. The file may reside on the program tape or a separate tape.

NOTES:

Option #7 - Renumber lines

The numbers used are for editing purposes only, they do not affect the numbers to the left of the line.

Option #6 - Search and Replace Characters

List the part of the file and note the line number where the first replacement should occur before doing this routine. This will save time and cut down the possibility of error.

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

EXAMPLES

```
1 = MAKE TEXT FROM KEYBOARD OR GET TEXT FROM FILE
2 = LIST LINES OF TEXT
3 = DELETE LINES OF TEXT
4 = INSERT LINES
5 = SWAP LINES
6 = SEARCH AND REPLACE CHARACTERS
7 = RENUMBER LINES
8 = SAVE ON TAPE
9 = LIST TEXT TO PRINTER
10= QUIT

WHICH ACTION?
1
ENTER INPUT DEVICE (KEYBOARD=31) (TAPE=33)
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
ENTER TEXT FROM KEYBOARD, WHEN FINISHED ENTER '$'
 1 THIS IS A TEXT FILE
 2 THE NUMBER AT THE BEGINNING IS ONLY A REFERENCE,
 3    IT IS NOT PART OF THE TEXT.
 4 With a regular text file like this one, do not use
 5   'RENUMBER' (Option #■).
 6 Opti■n #7 takes the first group of charters in each
Option #7 takes the first group of charac          ters in each
Option #7 takes the first group of characters in each line,
 7    and replaces it with a number.
 8 In line 6, you can see how the EXPAND, COMPRESS LOOKS.
 9 In lines 5 & 6, you can see the ■■■ RUBOUTS.
 10 THAT'S ENOUGH FOR NOW.  !"#$%&'()*+,-./0123456789:;<=>?@
 11 $
THE FILE IS 10 LINES LONG.
PRESS RETURN TO GO ON.
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
   2 = LIST LINES OF TEXT

ALL (1)  OR JUST PART OF IT (2)?
1

******************** TOTAL TEXT FILE **********************
 1 THIS IS A TEXT FILE
 2 THE NUMBER AT THE BEGINNING IS ONLY A REFERENCE,
 3   IT IS NOT PART OF THE TEXT.
 4 With a regular text file like this one, do not use
 5   'RENUMBER' (Option #7).
 6 Option #7 takes the first group of characters in each line,
 7    and replaces it with a number.
 8 In line 6, you can see how the EXPAND, COMPRESS LOOKS.
 9 In lines 5 & 6, you can see the     RUBOUTS.
 10 THAT'S ENOUGH FOR NOW.  !"#$%&'()*=-:}]{[~↑?/)<,.`@+;
PRESS RETURN TO GO ON.


ALL (1)  OR JUST PART OF IT (2)?
2
LIST WHAT RANGE OF LINES? 3,7
 3   IT IS NOT PART OF THE TEXT.
 4 With a regular text file like this one, do not use
 5   'RENUMBER' (Option #7).
 6 Option #7 takes the first group of characters in each line,
 7    and replaces it with a number.
PRESS RETURN TO GO ON.
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
 3 = DELETE LINES OF TEXT


1 = MAKE TEXT FROM KEYBOARD OR GET TEXT FROM FILE
2 = LIST LINES OF TEXT
3 = DELETE LINES OF TEXT
4 = INSERT LINES
5 = SWAP LINES
6 = SEARCH AND REPLACE CHARACTERS
7 = RENUMBER LINES
8 = SAVE ON TAPE
9 = LIST TEXT TO PRINTER
10= QUIT


WHICH ACTION?
3
DELETE LINES, FIRST LINE THROUGH LAST LINE TO DELETE
3,3
NUMBER OF LINES IN PROGRAM IS 9.
DELETION COMPLETED
PRESS RETURN TO GO ON.
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
4 = INSERT LINES


 1 = MAKE TEXT FROM KEYBOARD OR GET TEXT FROM FILE
 2 = LIST LINES OF TEXT
 3 = DELETE LINES OF TEXT
 4 = INSERT LINES
 5 = SWAP LINES
 6 = SEARCH AND REPLACE CHARACTERS
 7 = RENUMBER LINES
 8 = SAVE ON TAPE
 9 = LIST TEXT TO PRINTER
 10= QUIT


 WHICH ACTION?
 4
 INSERT BEFORE LINE  4




ENTER LINES TO BE INSERTED. WHEN DONE ENTER $
THE LINE JUST BEFORE IS:
 3 With a regular text file like this one, do not use
 4   IT IS NOT A PART OF THE TEXT.
 5 $
NUMBER OF LINES IN PROGRAM IS 10
INSERTION COMPLETED
PRESS RETURN TO GO ON.
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
5 = SWAP LINES

1 = MAKE TEXT FROM KEYBOARD OR GET TEXT FROM FILE
2 = LIST LINES OF TEXT
3 = DELETE LINES OF TEXT
4 = INSERT LINES
5 = SWAP LINES
6 = SEARCH AND REPLACE CHARACTERS
7 = RENUMBER LINES
8 = SAVE ON TAPE
9 = LIST TEXT TO PRINTER
10= QUIT


WHICH ACTION?
5
ENTER THE TWO LINE NUMBERS YOU WISH TO SWAP 8,9
SWAP COMPLETED
PRESS RETURN TO GO ON.
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

6 = SEARCH AND REPLACE CHARACTERS

```
TYPE '1' TO START AT THE BEGINNING, THE LINE# IF YOU KNOW IT
11
 11 REM *** THE 11 IS THE PROGRAM LINE NUMBER
HOW MANY TIMES DO YOU WANT THE REPLACEMENT? 3

 PLEASE TYPE IN THE PART TO BE REPLACED:
IF THIS IS NOT THE FIRST OCCURANCE, YOU'LL BLOW IT!
*

PLEASE TYPE IN THE REPLACEMENT OF THE ABOVE:
...

     ->REM *** THE 11 IS THE PROGRAM LINE NUMBER
     ->REM ...** THE 11 IS THE PROGRAM LINE NUMBER

     ->REM ...** THE 11 IS THE PROGRAM LINE NUMBER
     ->REM ......* THE 11 IS THE PROGRAM LINE NUMBER

     ->REM ......* THE 11 IS THE PROGRAM LINE NUMBER
     ->REM ......... THE 11 IS THE PROGRAM LINE NUMBER
REPLACEMENT COMPLETED
PRESS RETURN TO GO ON.
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
   7 = RENUMBER LINES


1 = MAKE TEXT FROM KEYBOARD OR GET TEXT FROM FILE
2 = LIST LINES OF TEXT
3 = DELETE LINES OF TEXT
4 = INSERT LINES
5 = SWAP LINES
6 = SEARCH AND REPLACE CHARACTERS
7 = RENUMBER LINES
8 = SAVE ON TAPE
9 = LIST TEXT TO PRINTER
10= QUIT


WHICH ACTION?
7
ENTER FIRST AND LAST LINES TO BE RENUMBERED (1,7) 1,11
ENTER NUMBER WITH WHICH TO START AND INCREMENT (10,10) 10,10
RENUMBERING COMPLETE
PRESS RETURN TO GO ON.
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
   8 = SAVE ON TAPE


 1 = MAKE TEXT FROM KEYBOARD OR GET TEXT FROM FILE
 2 = LIST LINES OF TEXT
 3 = DELETE LINES OF TEXT
 4 = INSERT LINES
 5 = SWAP LINES
 6 = SEARCH AND REPLACE CHARACTERS
 7 = RENUMBER LINES
 8 = SAVE ON TAPE
 9 = LIST TEXT TO PRINTER
 10= QUIT


 WHICH ACTION?
 8
 ENTER FILE NUMBER
 BE SURE YOU HAVE A FILE MARKED THERE!
 2
 YOUR TEXT/PROGRAM IS SAVED IN FILE #2.
 GOODBYE!
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

9 = LIST TEXT TO PRINTER

```
WHAT IS THE DEVICE NUMBER FOR THE PRINTER? 51
ALL (1)  OR JUST PART OF IT (2)?
1

********** TOTAL TEXT FILE **********
 1  2  3  4  5  6  7  8  9  10 PRESS RETURN TO GO ON.
```

```
10 IS A TEXT FILE
20 NUMBER AT THE BEGINNING IS ONLY A REFERENCE,
30 a regular text file like this one, do not use
40  IT IS NOT A PART OF THE TEXT.
50  'RENUMBER' (Option #7).
60 #7 takes the first group of characters in each line,
70   and replaces it with a number.
80 line 6, you can see how the EXPAND, COMPRESS LOOKS.
90 lines 5 & 6, you can see the     RUBOUTS.
100 ENOUGH FOR NOW.  !"#$%&'()*=-:}]{[~^?/><,.`@+;
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

DATA STORAGE

Vairable	Used to Store	Type
A	Length of a text line	Array
A$	Main text	String
B	Beginning character number in A$ of a text line	Array
B$	One line of input text	String
C$	Temporary string carrier	String
D1	Printer device number	Simple
H	Which action	Simple
H1	Temporary input	Simple
H$	Temporary string input	String
I	Temporary counter	Simple
I1	Temporary input	Simple
J	Temporary counter	Simple
J1	Temporary input	Simple
K	Temporary counter	Simple
L	Line number	Simple
L1	Temporary input	Simple
L2	Temporary input	Simple
M$	Temporary string input	String
N	Memory	Simple
P	Temporary counter	Simple
Q	Temporary counter	Simple
Q$	Temporary string carrier	String
R$	Temporary string carrier	String
T	Number of lines in a file	Simple
U1	Form of input	Simple
W$	Temporary string input	String
X	File number of data	Simple

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
4 GO TO 100
100 PAGE
110 PRINT "1 = MAKE TEXT FROM KEYBOARD OR GET TEXT FROM FILE"
120 PRINT "2 = LIST LINES OF TEXT"
130 PRINT "3 = DELETE LINES OF TEXT"
140 PRINT "4 = INSERT LINES"
150 PRINT "5 = SWAP LINES"
160 PRINT "6 = SEARCH AND REPLACE CHARACTERS"
170 PRINT "7 = RENUMBER LINES"
180 PRINT "8 = SAVE ON TAPE"
190 PRINT "9 = LIST TEXT TO PRINTER"
200 PRINT "10= QUIT"
210 PRINT "GJJWHICH ACTION?"
220 INPUT H
230 GO TO H OF 270,590,800,1020,1370,1540,2370,2080,590,2640

240 REM  *** TEXT EDITOR ***
250 REM      RE-WRITTEN 1/24/80  DENNIS WARD   =SID ACC.=
255 REM *** 51/00-8038/0 ***
260 REM
270 REM INPUT TEXT FILE - KILLS OLD TEXT, IF ANY

280 INIT
290 DIM B(300),A(300),A$(9000),B$(80)
300 A=0
310 A$=""
320 PRINT "ENTER INPUT DEVICE (KEYBOARD=31) (TAPE=33) ";
330 INPUT U1
340 PAGE
350 IF U1=31 THEN 400
360 PRINT "ENTER FILE NUMBER ";
370 INPUT X
380 FIND X
390 GO TO 410
400 PRINT "ENTER TEXT FROM KEYBOARD, WHEN FINISHED ENTER '$'"
410 FOR I=1 TO 2000
```

TITLE	ABSTRACT NO.
TEXT EDITOR	51/00-8038/0

```
420      IF U1=33 THEN 440
430      PRINT I;" ";
440      INPUT @U1:B$

450      REM     B() IS POSITION OF SEGMENT IN TEXT

460      B(I)=LEN(A$)+1
470      A$=REP(B$,B(I),0)

480      REM     A() IS LENGTH OF THE SEGMENT

490      A(I)=LEN(B$)
500      IF U1=31 THEN 530
510      ON EOF (0) THEN 550
520      GO TO 540
530      IF B$="$" THEN 550
540 NEXT I
550 T=I-2
551 IF U1=33 THEN 560
555 T=T+1

560 REM  THE ENTIRE TEXT IS A$

570 PRINT "THE FILE IS ";T;" LINES LONG."
580 GO TO 1340

590 REM     LISTING OF TEXT

600 PAGE
601 IF H<>9 THEN 610
602 PRINT "WHAT IS THE DEVICE NUMBER FOR THE PRINTER? ";
603 INPUT D1
610 PRINT "ALL (1)  OR JUST PART OF IT (2)?"
620 INPUT H1
630 IF H1=2 THEN 670
640 L1=1
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
650 PRINT "J******************** TOTAL TEXT FILE **********************"
660 GO TO 700
670 PRINT "LIST WHAT RANGE OF LINES? ";
680 INPUT L1,L2
690 IF L2<=T THEN 710
700 L2=T
710 FOR I=L1 TO L2
720     B$=SEG(A$,B(I),A(I))
730     PRINT I;" ";
740     IF H=9 THEN 770
750     PRINT B$
760     GO TO 780
770     PRINT @D1:B$
780 NEXT I
790 GO TO 1340

800 REM     DELETE LINES

810 PRINT "DELETE LINES, FIRST LINE THROUGH LAST LINE TO DELETE"
820 INPUT L1,L2
830 IF L2<=T THEN 850
840 L2=T
850 DELETE K
860 K=L2-L1+1
870 H=T
880 FOR J=L1 TO L2
890     FOR I=L1 TO T
900         A(H+1)=0
910         B(H+1)=LEN(A$)+1
920         A(I)=A(I+1)
930         B(I)=B(I+1)
940     NEXT I
950     H=H+1
960 NEXT J
970 T=T-K
980 PRINT "NUMBER OF LINES IN PROGRAM IS ";T;"."
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
990 GOSUB 1880
1000 PRINT "DELETION COMPLETED"
1010 GO TO 1340

1020 REM    INSERT LINES

1030 PRINT "INSERT BEFORE LINE  ";
1040 INPUT L1
1050 PAGE
1060 IF L1<=T THEN 1080
1070 L1=T+1
1080 PRINT "ENTER LINES TO BE INSERTED. WHEN DONE ENTER $"
1090 IF L1=1 THEN 1140
1100 PRINT "THE LINE JUST BEFORE IS:"
1110 B$=SEG(A$,B(L1-1),A(L1-1))
1120 PRINT L1-1;" ";B$
1130 GO TO 1150
1140 PRINT "THIS WILL BE THE FIRST LINE"
1150 PRINT L1;" ";
1160 INPUT B$
1170 IF B$="$" THEN 1310
1180 FOR I=T TO L1 STEP -1
1190     A(I+1)=A(I)
1200     B(I+1)=B(I)
1210 NEXT I
1220 B(L1)=LEN(A$)+1
1230 A(L1)=LEN(B$)
1240 A$=REP(B$,B(L1),0)
1250 L1=L1+1
1260 T=T+1
1270 IF MEMORY>900 AND LEN(A$)<8900 THEN 1130
1280 PRINT "YOU BETTER SAVE THIS NOW!!G"
1290 GO TO 1310
1300 GO TO 1130
1310 PRINT "NUMBER OF LINES IN PROGRAM IS ";T
1320 GOSUB 1880
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
1330 PRINT "INSERTION COMPLETED"
1340 PRINT "PRESS RETURN TO GO ON."
1350 INPUT H$
1360 GO TO 100

1370 REM     SWAP LINES ROUTINE

1380 PRINT "ENTER THE TWO LINE NUMBERS YOU WISH TO SWAP ";
1390 INPUT I1,J1
1400 IF I1<=T THEN 1420
1410 GO TO 1380
1420 IF J1<T THEN 1440
1430 J1=T
1440 DELETE K
1450 DIM K(T+1)
1460 K(I1)=A(I1)
1470 A(I1)=A(J1)
1480 A(J1)=K(I1)
1490 K(I1)=B(I1)
1500 B(I1)=B(J1)
1510 B(J1)=K(I1)
1520 PRINT "SWAP COMPLETED"
1530 GO TO 1340

1540 REM REPLACE ROUTINE

1550 PAGE
1560 PRI "TYPE '1' TO START AT THE BEGINNING, THE LINE# IF YOU KNOW IT"
1570 INPUT L1
1580 PRINT L1;" ";
1590 B$=SEG(A$,B(L1),A(L1))
1600 PRINT B$
1610 PRINT "HOW MANY TIMES DO YOU WANT THE REPLACEMENT? ";
1620 INPUT H
1630 PRINT "J PLEASE TYPE IN THE PART TO BE REPLACED:"
1640 PRINT "IF THIS IS NOT THE FIRST OCCURANCE, YOU'LL BLOW IT!"
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
1650 INPUT M$
1660 PRINT "JPLEASE TYPE IN THE REPLACEMENT OF THE ABOVE:"
1670 INPUT W$
1680 FOR I=1 TO H
1690     FOR L=L1 TO T
1700         B$=SEG(A$,B(L),A(L))
1710         P=POS(B$,M$,1)
1720         IF P=0 THEN 1770
1730         PRINT "J     ->";B$
1740         B$=REP(W$,P,LEN(M$))
1750         PRINT "      ->";B$
1760         GO TO 1780
1770     NEXT L
1780     IF L<=T THEN 1810
1790     PRINT "NOT FOUND!!G"
1800     GO TO 1870
1810     A(L)=LEN(B$)
1820     B(L)=LEN(A$)+1
1830     A$=REP(B$,B(L),0)
1840 NEXT I
1850 GOSUB 1880
1860 PRINT "REPLACEMENT COMPLETED"
1870 GO TO 1340

1880 REM     CLEAN FILE ROUTINE

1890 DELETE B$
1900 N=MEMORY
1910 DIM B$(INT(N-170))
1920 B$=""
1930 FOR I=1 TO T
1940     C$=SEG(A$,B(I),A(I))
1950     IF LEN(B$)+LEN(C$)>INT(N-170) THEN 2060
1960     B(I)=LEN(B$)+1
1970     A(I)=LEN(C$)
1980     B$=REP(C$,B(I),0)
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
1990 NEXT I
2000 DELETE A$
2010 N=MEMORY
2020 DIM A$(INT(N-170))
2030 A$=B$
2040 DELETE B$
2050 RETURN
2060 PRINT "NOT COMPLETED - WE'RE NEARLY OUT OF MEMORY!G"
2070 GO TO 1340

2080 REM     SAVE FILE ROUTINE

2090 DELETE B$
2100 N=MEMORY
2110 DELETE C$
2120 DIM C$(80)
2130 DIM B$(LEN(A$)+2*T)
2140 B$=""
2150 R$=CHR(13)
2160 FOR I=1 TO T
2170     C$=SEG(A$,B(I),A(I))
2180     C$=C$&R$
2190     B(I)=LEN(B$)+1
2200     B$=REP(C$,B(I),0)
2210 NEXT I
2220 DELETE A$
2230 N=MEMORY
2240 DIM A$(INT(N-170))
2250 A$=B$
2260 IF LEN(A$)=>LEN(B$) THEN 2280
2270 PRINT "GSOME TEXT WAS LOST - NEARLY OUT OF MEMORY!!G"
2280 DELETE B$
2290 PRINT "ENTER FILE NUMBER"
2300 PRINT "GBE SURE YOU HAVE A FILE MARKED THERE!"
2310 INPUT X
2320 FIND X
```

TITLE	ABSTRACT NO:
TEXT EDITOR	51/00-8038/0

```
2330 PRINT @33:A$
2340 PRINT "YOUR TEXT/PROGRAM IS SAVED IN FILE #";X;"."
2350 PRINT "GOODBYE!"
2360 END

2370 REM     RENUMBER ROUTINE

2380 PRINT "ENTER FIRST AND LAST LINES TO BE RENUMBERED (1,7) ";
2390 INPUT H1,U1
2400 PRINT "ENTER NUMBER WITH WHICH TO START AND INCREMENT (10,10) ";
2410 INPUT H,I
2420 Q=H-I
2430 IF U1<=T THEN 2450
2440 U1=T
2450 FOR L=H1 TO U1
2460     IF LEN(A$)<8900 AND MEMORY>900 THEN 2500
2470     PRINT "JWE ARE ALMOST OUT OF MEMORY! G"
2480     PRINT "RENUMBERING NOT COMPLETED!J"
2490     GO TO 2122
2500     B$=SEG(A$,B(L),A(L))
2510     P=POS(B$," ",1)
2520     Q=Q+I
2530     Q$=STR(Q)
2540     Q$=Q$&" "
2550     B$=REP(Q$,1,P)
2560     B$=SEG(B$,2,72)
2570     B(L)=LEN(A$)+1
2580     A(L)=LEN(B$)
2590     A$=REP(B$,B(L),0)
2600 NEXT L
2610 GOSUB 1880
2620 PRINT "RENUMBERING COMPLETE"
2630 GO TO 1340
2640 END
```

TEKTRONIX

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
VARIABLE FONT PRESENTATION AID	51/00-8040/0
ORIGINAL DATE: April 1980 / REVISION DATE:	MEMORY REQUIREMENT: 32K
AUTHOR: William R. Sebra, U.S. AMSAA, Aberdeen Proving Ground	PERIPHERALS: Optional - 4662 Plotter

ABSTRACT

Statements: 331

Files: 1 ASCII Program
8 Binary Data
(First 9 files on tape or reassign file # variables)

The program produces a page of text, oriented in either a horizontal or vertical format, of high quality drawn characters suitable for projection as a presentation aid. Four fonts are available in various sizes for the text:

Roman

Roman Italic

Gothic

Script

Each line of text can have a different character size and a different margin selection (centered, right or left justified). The text is automatically centered on the page by the program. Output may be to the screen or the 4662 Plotter.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

DATA TAPE STRUCTURE

File 1 - Program file (ASCII)
File 2 - Data file (binary numeric) for Font 1
File 3 - Data file (binary string) for Font 1
File 4 - Data file (binary numeric) for Font 2
File 5 - Data file (binary string) for Font 2
File 6 - Data file (binary numeric) for Font 3
File 7 - Data file (binary string) for Font 3
File 8 - Data file (binary numeric) for Font 4
File 8 - Data file (binary string) for Font 4

PROGRAM LOADING

Press AUTOLOAD if the program is located on the first files of the tape, otherwise:

FIND # (file number of program)

OLD

RUN or Press User-Definable Key #1

The program must be executed once before the parameters may be changed using the User-Definable Keys.

OPERATING INSTRUCTIONS

Once the program has been started (or restarted using User-Definable Key #1), it will step through the sequence of operations until the text is drawn. After the initial draw, the User-Definable Keys may be used.

Device selection is made by entering either a 32 (screen) or 1 (plotter) in response to the prompt.

Font selection is made by entering the number corresponding to the font desired: 1 = Roman, 2 = Roman Italic, 3 = Gothic, 4 = Script.

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

Format selection is made by entering 1 for horizontal or 2 for vertical. Horizontal format draws each line of text parallel with the long dimension of the paper, vertical format draws each line parallel with the short dimension of the page. When using the plotter the page should always be set with the long dimension in the X direction.

Text input is done in the following way. The screen is paged by the program and a heading is written at the top of the page.

Margin Size Text

For each input, the blinking prompt appears in the appropriate section.

Margin - Margin is entered as one of the following:

L = left justified
R = right justified
C = centered

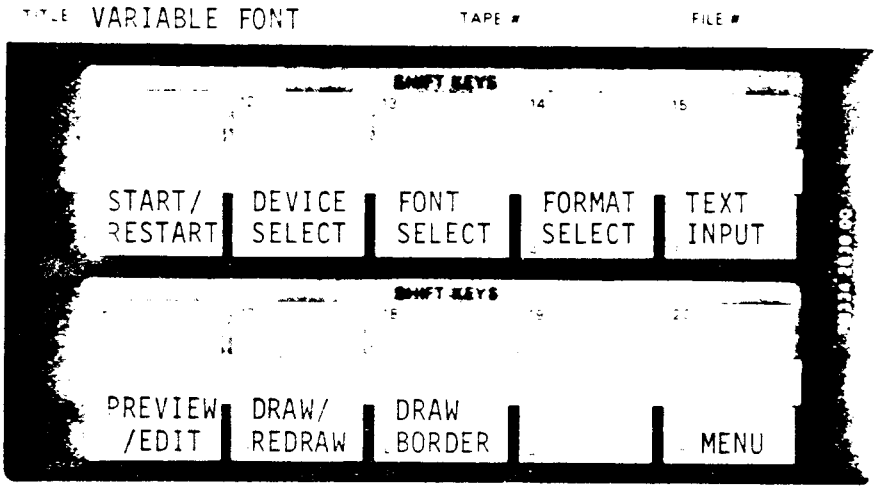
After a margin definition is input, subsequent entries need only a carriage return to be entered unless a change is to be made.

Size - Size is entered as lines per page, but for versatility, non-integers may be entered. After a size is input, subsequent entries need only a carriage return to be entered unless a change is to be made.

Text - Text is entered one line at a time. At least one character must be entered, even if it is only a space. Text input is terminated when a carriage return is entered as the first character of a line.

After the text has been entered, the program pages the screen and prints out the text, with appropriate adjustments for margins so that the screen reflects what will be drawn. At this time, the user has the opportunity to edit a line by deleting it, inserting a new line or changing margin or size definitions.

TITLE	ABSTRACT NUMBER
VARIABLE FONT PRESENTATION AID	51/00-8040/0



UDK #	Function
1	Start/restart runs the program from the beginning
2	Device select 32 = screen, 1 = plotter
3	Font select 1 = Roman, 2 = Roman Italic, 3 = Gothic, 4 = Script
4	Format select 1 = vertical, 2 = horizontal
5	Text input
6	Preview/edit the current text on the screen with normal alpha-numeric characters with adjustments to reflect the effect of the effect of the margin controls. The user may then edit the text by deleting lines, inserting lines, or changing margin definitions for a line.
7	Draw/redraw draws the current text
8	Draw border draws a full page border around the text
10	Menu displays the list of User-Definable Keys and their functions

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-8040/0

Features of Variable Font Presentation Aid

1. Four different fonts.
2. Horizontal or vertical format.
3. Margins may be

right justified

centered

left justified
4. Change of size

from line

to line

5. Text preview/edit.
6. Redraw with 1-5 changed.

Features of Variable Font Presentation Aid

1. Four different fonts.
2. Horizontal or vertical format.
3. Margins may be

right justified

centered

left justified

4. Change of size

from line

to line

5. Text preview/edit.
6. Redraw with 1–5 changed.

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO.

51/00-8040/0

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

Roman

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm
Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz
0123456789 !"$&'()*=+;:?,./

Roman Italic

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm
Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz
0123456789 !"$&'()=+;:?,./*

Gothic

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm
Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz
0123456789 !"$&'()*=+;:?,./

Script

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm
Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz
0123456789 !"$&'()=+;:?,./*

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO

51/00-8040/0

INTERNAL DATA STORAGE

Variable	Used to Store	Type
A$	Temporary variable	String (1000)
B	Coordinate of line segment of current line to be drawn (GDU's - relative)	Simple
B1	Coordinate of line segment of current line to be drawn (GDU's - relative)	Simple
B$	Intermediate variable for character currently being drawn	String
C	Number of lines of text plus one	Simple
C1	Cumlative number of characters, and width of line (GDU's)	Array (80,2)
D	Position of line segment being drawn, and height and width of character being drawn (GDU's)	Array (100,3)
E$	ASCII representation of current line	String
F9	Temporary variable	Simple
F$	Temporary variable	String
H	Spacing between lines of text	Array (80)
H1	Starting position of current line within text	Simple
H2	Character offset position in opposite direction from L1 (user units)	Simple
H$	ASCII equivalent points for current font character set	String (7700)
I	Temporary variable	Simple
K1	Maximum width of a line of text	Simple
K2	Constant for conversion of character size from lines per page to user units	Simple
K3	Character size of current line in user units	Simple
K4	Maximum width of line of text	Simple
L	Character size of a line in lines per page	Array (80)
L1	Character offset position in opposite direction from H2 (user units)	Simple
L8	Temporary variable	Simple
L9	Temporary variable	Simple

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

Variable	Used to Store	Type
M8	Temporary variable	Simple
M9	Margin parameter for a line	Array (80)
M$	Temporary variable	String
N	Temporary variable	Simple
N1	Temporary variable	Simple
N$	Temporary variable	String
O	Read data file loop	Simple
P	Temporary variable	Simple
Q	Temporary variable	Simple
R$	Text	String (2000)
S	Input insert line	Simple
T	Temporary variable	Simple
T$	Temporary variable	String (1000)
U	Temporary variable	Simple
W9	Long dimension of page (130 for screen, 150 for plotter)	Simple
Z	Device # (for output)	Simple
Z1	Current margin parameter	Simple

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
4 GO TO 100
8 GO TO 170
9 END
10 GO TO 170
12 GO TO 400
13 END
16 GO TO 650
17 END
20 GO TO 760
21 END
24 GO TO 2030
25 END
28 GO TO 1250
29 END
32 GO TO 3180
40 GO TO 280
100 INIT
110 DIM D(100,3),H$(7700)
120 GOSUB 170
130 GOSUB 400
140 GOSUB 650
150 GOSUB 760
160 GO TO 1250
170 PAGE

180 REM *** 51/00-8040/0 VARIABLE FONT PRESENTATION AID  ***
190 REM
200 REM ***                DEVICE SELECTION                ***

210 PRINT USING "///""Select Device(32=Screen,1=Plotter) -""S":
220 INPUT Z
230 IF Z<>32 AND Z<>1 THEN 170
240 W9=130*(Z=32)+150*(Z=1)
250 WINDOW 0,W9,0,100
260 VIEWPORT 0,W9,0,100
270 RETURN
```

TITLE	ABSTRACT NO
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
280 PAGE

290 REM
300 REM       **              PRINT MENU                    **
310 REM

320 PRINT USING "32T""User Keys""":
330 PRINT USING "///":
340 PRINT USING """Key #1 - Start/Restart""/""Key #2 - Device select""":
350 PRINT USING """Key #3 - Font select""/""Key #4 - Format select""":
360 PRI USI """Key #5 - Text input""/""Key #6 - Text preview/Edit""":
370 PRINT USING """Key #7 - Draw/Redraw""/""Key #8 - Draw border""":
380 PRINT USING """Key #10 - Display this MENU""":
390 END
400 PAGE

410 REM
420 REM       **              FONT SELECTION                **
430 REM

440 PRI "JJJJ  FONT:  1=ROMAN  2=ROMAN ITALIC   3=GOTHIC   4=SCRIPT  ";
450 INPUT F$
460 F$="0"&F$
470 Q=VAL(F$)
480 IF Q<1 OR Q>4 THEN 280
490 FIND Q*2

500 REM
510 REM       **           READ FONT DATA FROM TAPE         **
520 REM

530 FOR P=1 TO 100
540     FOR O=1 TO 3
550         IF TYP(0)=1 THEN 590
560         READ @33:D(P,O)
570     NEXT O
```

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
580 NEXT P
590 P=P-1
600 FIND Q*2+1
610 DIM D(P,3)
620 DIM H$(D(P,1))
630 READ @33:H$
640 RETURN
650 PAGE

660 REM
670 REM       **          FORMAT SELECTION              **
680 REM

690 PRINT USING "///""Vertical(1) or Horizontal(2) ? ""S":
700 INPUT F9
710 IF F9<0 OR F9>2 THEN 690
720 K4=100*(F9=1)+W9*(F9=2)
730 K1=100*(F9=2)+W9*(F9=1)
740 K2=K1/30
750 RETURN
760 PAGE

770 REM
780 REM       **              TEXT INPUT              **
790 REM

800 C=1
810 DIM C1(80,2),M9(80),H(80),L(80)
820 M9=0
830 H=0
840 L=0
850 PRINT USING """Margin Size""29T""Text""":
860 SET CASE
870 DELETE R$
880 DIM R$(2000),A$(1000),T$(1000)
890 R$=""
```

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
900 PRINT "  ";

910 REM
920 REM       **          MARGIN DEFINITION INPUT          **
930 REM

940 INPUT M$
950 IF C=1 AND LEN(M$)=0 THEN 900
960 IF LEN(M$)=0 AND C>1 THEN 990
970 M9(C)=(M$="L")*1+(M$="R")*2+(M$="C")*3
980 GO TO 1000
990 M9(C)=M9(C-1)
1000 PRINT USING """K""9IS":

1010 REM
1020 REM       **                SIZE INPUT               **
1030 REM

1040 INPUT M$
1050 IF LEN(M$)=0 AND C=1 THEN 1000
1060 IF LEN(M$)>0 THEN 1090
1070 L(C)=L(C-1)
1080 GO TO 1110
1090 M$="0"&M$
1100 L(C)=VAL(M$)
1110 PRINT USING """K""13IS":

1120 REM
1130 REM       **                TEXT INPUT               **
1140 REM

1150 INPUT A$
1160 IF LEN(A$)=0 THEN 1210
1170 C1(C,1)=LEN(R$)
1180 C=C+1
1190 R$=R$&A$
```

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
1200 GO TO 900
1210 C1(C,1)=LEN(R$)
1220 PRINT "End of text....."
1230 GO TO 2030
1240 RETURN
1250 PAGE

1260 REM
1270 REM       **              DRAW TEXT                    **
1280 REM

1290 H1=0
1300 FOR I=1 TO C-1
1310     C1(I,2)=0
1320 NEXT I
1330 FOR I=1 TO C-1
1340     K3=K2/L(I)
1350     FOR P=C1(I,1)+1 TO C1(I+1,1)
1360         B$=SEG(R$,P,1)
1370         B=ASC(B$)-31
1380         C1(I,2)=C1(I,2)+D(B,3)-D(B,2)
1390     NEXT P
1400     H(I)=(12+18)*K3
1410     H1=H1+H(I)
1420 NEXT I
1430 H1=H1/2+W9/2*(F9=1)+50*(F9=2)
1440 H2=H1+H(1)/2
1450 FOR I=1 TO C-1
1460     K3=K2/L(I)
1470     H2=H2-(H(I-(I>1))+H(I))/2
1480     Z1=M9(I)
1490     GO TO Z1 OF 1530,1580,1630

1500     REM
1510     REM       **            LEFT JUSTIFY                 **
1520     REM
```

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
1530     L1=0
1540     GO TO 1640

1550     REM
1560     REM       **            RIGHT JUSTIFY                    **
1570     REM

1580     L1=K4-K3*C1(I,2)
1590     GO TO 1640

1600     REM
1610     REM       **                CENTER                       **
1620     REM

1630     L1=(K4-K3*C1(I,2))/2
1640     FOR P=C1(I,1)+1 TO C1(I+1,1)
1650         F$=SEG(R$,P,1)
1660         N=ASC(F$)-31
1670         L1=L1-D(N,2)*K3
1680         GOSUB 1730
1690         L1=L1+D(N,3)*K3
1700     NEXT P
1710 NEXT I
1720 END

1730 REM
1740 REM       **          CHARACTER DRAW ROUTINE             **
1750 REM

1760 T=1
1770 IF D(N,1)=D(N+1,1) THEN 2020
1780 U=1
1790 E$=SEG(H$,U+D(N,1),1)
1800 B=ASC(E$)
1810 E$=SEG(H$,U+1+D(N,1),1)
```

TITLE	ABSTRACT NO
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
1820 B1=ASC(E$)
1830 B=80-B
1840 B1=80-B1
1850 IF B=47 AND B1=47 THEN 2020
1860 IF B=-46 AND B1=-46 THEN 1990
1870 IF T=1 THEN 1930
1880 IF F9=1 THEN 1910
1890 DRAW @Z:B*K3+L1,B1*K3+H2
1900 GO TO 2000
1910 DRAW @Z:B1*K3+H2,-B*K3-L1+100
1920 GO TO 2000
1930 IF F9=1 THEN 1960
1940 MOVE @Z:B*K3+L1,B1*K3+H2
1950 GO TO 1970
1960 MOVE @Z:B1*K3+H2,-B*K3-L1+100
1970 T=0
1980 GO TO 2000
1990 T=1
2000 U=U+2
2010 GO TO 1790
2020 RETURN
2030 PAGE

2040 REM
2050 REM      **              EDIT TEXT                    **
2060 REM

2070 FOR I=1 TO C-1
2080     N1=C1(I+1,1)-C1(I,1)
2090     M$="T"
2100     N$="4"
2110     GO TO M9(I) OF 2260,2170,2220

2120     REM
2130     REM      **              PRINT A LINE                    **
2140     REM
```

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
2150      REM       **              RIGHT JUSTIFIED                  **
2160      REM

2170      N$=STR(73-N1)
2180      GO TO 2260

2190      REM
2200      REM       **                 CENTERED                      **
2210      REM

2220      N$=STR(INT((77-N1)/2))

2230      REM
2240      REM       **              LEFT JUSTIFIED                   **
2250      REM

2260      M$=N$&M$
2270      M$="2D1X"&M$
2280      M$=M$&"S"
2290      PRINT USING M$:I
2300      M$=SEG(R$,C1(I,1)+1,N1)
2310      PRINT M$
2320 NEXT I
2330 PRINT "To edit a line, enter 1, else RETURN ";
2340 INPUT M$
2350 IF LEN(M$)<>0 THEN 2370
2360 RETURN
2370 PRINT "Line # to be edited ?";
2380 INPUT L9
2390 PAGE

2400 REM       **              PRINT LINE TO BE EDITED       **

2410 PRINT "LINE # ";L9
2420 N1=C1(L9+1,1)-C1(L9,1)
2430 N$=SEG(R$,C1(L9,1)+1,N1)
```

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
2440 GOSUB 3030
2450 PRINT "Delete(1), insert(2), change size/margin(3), or exit(CR)? ";
2460 INPUT M$
2470 IF LEN(M$)=0 THEN 2030
2480 IF ASC(M$)<49 OR ASC(M$)>51 THEN 2450
2490 L8=VAL(M$)
2500 GO TO L8 OF 2550,2630,2930
2510 GO TO 2030

2520 REM
2530 REM      **              DELETE A LINE                  **
2540 REM

2550 C=C-1
2560 R$=REP("",C1(L9,1)+1,N1)
2570 FOR I=L9 TO C
2580     M9(I)=M9(I+1)
2590     L(I)=L(I+1)
2600     C1(I,1)=C1(I+1,1)-N1
2610 NEXT I
2620 GO TO 2030
2630 PAGE

2640 REM
2650 REM      **              INSERT A LINE                  **
2660 REM

2670 GOSUB 3030
2680 INPUT M$
2690 IF LEN(M$)=0 OR (M$<>"L" AND M$<>"R" AND M$<>"C") THEN 2630
2700 M8=(M$="L")*1+(M$="R")*2+(M$="C")*3
2710 PRINT USING """K""9TS":
2720 INPUT S
2730 PRINT USING """K""13TS":
2740 INPUT A$
2750 N1=LEN(A$)
```

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-80-0/0

```
2760 IF N1=0 THEN 2630
2770 T$=SEG(R$,0,C1(L9+1,1))
2780 T$=T$&A$
2790 A$=SEG(R$,C1(L9+1,1)+1,C1(C,1))
2800 R$=T$&A$
2810 C=C+1
2820 FOR I=C-1 TO L9+1 STEP -1
2830     M9(I)=M9(I-1)
2840     L(I)=L(I-1)
2850     C1(I+1,1)=C1(I,1)+N1
2860 NEXT I
2870 M9(L9+1)=M8
2880 L(L9+1)=S
2890 GO TO 2030

2900 REM
2910 REM      **          CHANGE MARGIN, SIZE          **
2920 REM

2930 PRINT "Margin(L,R,C) ? ";
2940 INPUT M$
2950 IF M$<>"L" AND M$<>"R" AND M$<>"C" THEN 2930
2960 IF LEN(M$)=0 THEN 2930
2970 M9(L9)=(M$="L")*1+(M$="R")*2+(M$="C")*3
2980 PRINT "Size ? ";
2990 INPUT M$
3000 IF LEN(M$)=0 THEN 2980
3010 L(L9)=VAL(M$)
3020 GO TO 2030
3030 GO TO M9(L9) OF 3040,3060,3080
3040 M$="L"
3050 GO TO 3120
3060 M$="R"
3070 GO TO 3120
3080 M$="C"
```

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
3090 REM
3100 REM      **          PRINT TEXT HEADER ROUTINE          **
3110 REM

3120 PRINT USING """Margin Size""29T""Text""":
3130 PRINT USING "3T1A9TFD.D29T43A":M$,L(L9),N$
3140 RETURN

3150 REM
3160 REM      **             BORDER DRAW ROUTINE             **
3170 REM

3180 MOVE @Z:0,0
3190 DRAW @Z:W9,0
3200 DRAW @Z:W9,100
3210 DRAW @Z:0,100
3220 DRAW @Z:0,0
3230 HOME @Z:
3240 END
```

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-8040/0

ROMAN

0	-8	8
0	-5	6
68	-9	9
11[illegible]	0	0
11[illegible]	-10	10
[illegible]	0	0
[illegible]	-13	13
3[illegible]	-4	5
[illegible]	-7	7
448	-7	7
50[illegible]	-8	8
580	-1[illegible]	13
6[illegible]	-5	6
660	-1[illegible]	13
676	-5	5
708	-11	1[illegible]
7[illegible]	-10	10
8[illegible]	-10	10
880	-10	10
1004	-10	10
1156	-10	10
1[illegible]	-10	10
13[illegible]	-10	10
1460	-10	10
1546	-10	10
1704	-10	10
184[illegible]	-5	6
1906	-5	6
1986	-5	6
2034	-1[illegible]	13
[illegible]066	-5	6
[illegible]114	-9	10

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

2218	0	0
2218	-10	10
2290	-11	11
2446	-11	10
2520	-11	11
2644	-11	10
2810	-11	9
2950	-11	12
3070	-12	12
3234	-6	6
3310	-8	8
3400	-12	10
3538	-9	9
3642	-13	13
3768	-12	12
3846	-11	11
3954	-11	11
4072	-11	11
4226	-11	11
4386	-10	10
4474	-10	10
4588	-12	12
4678	-10	10
4746	-12	12
4856	-10	10
4964	-11	11
5060	-10	10
5142	0	0
5142	0	0
5142	0	0
5142	0	0
5142	0	0
5142	0	0
5142	-9	11
5250	-11	10
5344	-10	9
5412	-10	11
5516	-10	9
5598	-7	7
5678	-9	11
5856	-11	12
5986	-6	6
6072	-7	6
6154	-11	11
6276	-6	6
6338	-17	17
6536	-11	12
6666	-10	10
6758	-11	10
6878	-10	10
6988	-9	8
7074	-8	9
7160	-7	8
7204	-11	12
7290	-9	9
7352	-12	12
7442	-10	10
7538	-10	9
7618	-9	9
7700	0	0
7700	0	0
7700	0	0
7700	0	0
7700	0	0

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
H$
PDQEQGPO~~PDPROR~~PDODOR~~ODNENGOO~~PUQWQXPYOYNXNWOUPU~~PWPXOXOWPW!!TDUE
UK~~TEUK~~TDSEUK~~KDLELK~~KELK~~KDJELK!!R@RJ~~N@NJ~~JIJHKHKJIJIHJFKENDRD
UEWGHJULSNMPKQJSJUKX~~UJULSMMOKPJR~~UEUGUIUK9LMNJPIRIUJWKXNYRYUXUWWUWSUS
UUUUUT!!GMGLHLHNFNFLGKHKILJNLSNUPXRYUYXXYUYSXQRMPKOIOGPERDTEUGUJTMRPNUKX
IYGYFWFU~~WXXUXSWQUP~~PKOG~~OIPE~~TEUI~~TLRONTKWIX~~TYUXWUWSUQRM~~UGTKQO
MTJWHXGXFW!!ODPEPK~~OEPK~~ODNEPK!!M@OBQESITNTRSWQEO↑M`~~QFRISMSSRWQZ~~OB
PDQGRMRSQYP\O↑!!S@QBOEMILNLRMWOEQ↑S`~~OFNIMMMSNWOZ~~QBPDOGNMNSOYP\Q↑!!PD
QEOOPP~~PDPP~~PDOEQOPP~~UGTGLMKM~~UGUHKLKM~~KGLGTMUM~~KGUM~~KGKHUL
UM!!PGPXOX~~PGOGOX~~XOGOGP~~XOXPGP!!NXOYPYQXQWPUOUNWNZO\QJ~~PWPXOXOWPW~~
OYNZ~~NXO\!!XOGOGP~~XOXPGP!!PUQWQXPYOYNXNWOUPU~~PWPXOXOWPW!!G@Y`X`~~G@F@
X`!!QDTEUHWMWPUUTXQYOYLXJUIPIMJHLEODQD~~TFUHULVQUUTW~~LWKUJQJLKHLF~~QDSE
TGULUQTUSXQY~~OYMXLVKQKLLGMEOD!!QFQY~~PFPX~~ODOY~~ODRGTH~~UYKY~~QXSY~~QW
RY~~OWNY~~OXMY!!UHUIUIUHUH~~UGUGTHTIUJUJWIWHUFUERDNDKEJFIHIJJLMNRPTQUSWU
WY~~KFJHJJKL~~NDLEKHKJLLNNRP~~WWUUTUOWKWIU~~TUOXKXJW~~TUOYKYJXIUIT!!UHUI
UIUHUH~~UGUGTHTIUJUJWIWHUFUERDNDKEJGJJKLNM~~LEKGKJLL~~ODMELGLJMLOM~~QMNM
LHJPIRIUJWKXNYRYUXUWWUWTUSUSTTTUUUUU~~KPJRJUKW~~OMMNLOKRKULXNY~~UTUUUUUT
VT!!OGOY~~NFNX~~MDMY~~MDXSHS~~RYJY~~OXQY~~OWPY~~MWLY~~MXKY!!UDWNULRKOKLL
JNIQISJULXOYRYUXUWWUWTUSUSTTTUUUUU~~KNJPJTKU~~OKMLLMKPKTLWMXOY~~UTUUUUUT
VT~~UDKD~~UEME~~UFQFMEKD!!LGLHKHKGLG~~KFLFMGMHLIKIJHJGKEMDPDSEUGUIWMWSUU
TXQYOYLXJUISIRJOLMOLQLSMTNUP~~TGUIUMUSUUTW~~KUJTJQKO~~PDRESFTHULUSTUSXQY
~~OYMXLWKTKQLNMMOL!!WDWJ~~IDIGJJNOOQPUPY~~OPPRQUQY~~JJOOQRRURYPY~~WHUFTD
RDMGKGJFID~~UFTEREPF~~WHUGTFRFMG!!RDUEUGUJULRMNMKLJJJGKENDRD~~TEUGUJTL~~
LLKJKGLE~~RDSETGTJSLRM~~NMMLLJLGMEND~~RMUNUOWQWUUWUXRYNYKXJWIUIQJOKNNM~~
UOUQUUUW~~KWJUJQKO~~RMTNUQUUTXRY~~NYLXKUKQLNNM!!UUUUTUTUUU~~KMLOMPOQQQTP
UNWKWJUGTEQDODLEJGIJIPJTKUMXPYSYUXUVUUUTTTSUSUTWUW~~UNULVIUG~~LFKGJJJPKT
LU~~QQSPTOULUITFSEQD~~ODMELGKJKQLUMWNXPY!!PKQLQMPNONNMNLOKPK~~PLPMOMOLPL
~~PUQWQXPYOYNXNWOUPU~~PWPXOXOWPW!!PKQLQMPNONNMNLOKPK~~PLPMOMOLPL~~NXOYPY
QXQWPUOUNWNZO\QJ~~PWPXOXOWPW~~OYNZ~~NXO\!!NDPEQGQJPKOKNJNIOHPHQI~~PIPJOJ
OIPI~~PEQI~~QGPH!!XKGKGL~~XKXLGL~~XSGSGT~~XSXTGT!!NFOGPGQFQEPDODNENHOJQK
~~PEPFOFOEPE~~OGNH~~NFOJ!!UIUHTHTJUJUHUFTERDNDKEJFIHIJJLKMOO~~KFJGJKKL~~
NDLEKGKKLMMN~~POPRORQOPO~~PUQWQXPYOYNXNWOUPU~~PWPXOXOWPW!!PDWX~~QGKY~~PG
JY~~PDIY~~USLS~~YYSY~~NYGY~~WXXY~~KXMY~~KWLY~~JWHY!!UDUY~~UEUX~~TD
TY~~YDMDJEIFHHJILJMMN~~JFIHIJJL~~MDKEJGJKKMMN~~TNMNJOIPHRHUIWJXMYYY~~JP
IRIUJW~~MNKOJQJUKXMY~~XDUE~~WDUF~~SDTF~~RDTE~~UXXY~~UWWY~~TWSY~~TXRY!!JG
```

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
IDIJJGLENDQDTEUGWIXLXQWTVUTXQYNYLXJUIT~~UGUIWLWQUTUV~~QDSEUHULVQUUSXQY!!
UDUY~~UEUX~~TDTY~~YDODLEJGIIHLHQITJULXOYYY~~KGJIILIQJTKU~~ODMEKHJLJQKUMX
OY~~XDUE~~WDUF~~SDTF~~RDTE~~UVXXY~~UWWY~~TWSY~~TXRY!!UDUY~~UEUX~~TDTY~~YD
IDIJ~~TNNN~~NJNR~~YYIYIS~~XDUE~~WDUF~~SDTF~~RDTE~~NDIE~~LDIF~~KDIG~~JDIJ
~~NJONNR~~NLPNNP~~NMRNNO~~UVXXY~~UWWY~~TWSY~~TXRY~~NYIX~~LYIW~~KYIU~~JYIS
!!UDUY~~UEUX~~TDTY~~YDIDIJ~~TNNN~~NJNR~~YYQY~~XDUE~~WDUF~~SDTF~~RDTE~~ND
IE~~LDIF~~KDIG~~JDIJ~~NJONNR~~NLPNNP~~NMRNNO~~UVXXY~~UWWY~~TWSY~~TXRY!!JG
IDIJJGLENDQDTEUGWIXLXQWTUVTXQYNYLXJXIYIQ~~UGUIWLWQUTUV~~QDSEUHULVQUUSXQY
~~JRJW~~KQKWLX~~NQFQ~~MQKR~~LQKS~~HQIS~~GQIR!!WDWY~~VEUX~~UDUY~~KDKY~~JE
JX~~IDIY~~ZDRD~~NDFD~~UNKN~~ZYRY~~NYFY~~YDWE~~XDWF~~TDUF~~SDUE~~MDKE~~LD
KF~~HDIF~~GDIE~~WXYY~~WWXY~~UWTY~~UXSY~~KXMY~~KWLY~~IWHY~~IXGY!!QDQY~~PE
PX~~ODOY~~TDLD~~TYLY~~SDQE~~RDQF~~NDOF~~MDOE~~QXSY~~QWRY~~OWNY~~OXMY!!OD
OUPXQY~~MENUOX~~MDMUNXQYSYUXVVUTUSTSSTSUTUVU~~UTUUTUTTUT~~RDJD~~QDOE~~PD
OF~~LDMF~~KDME!!WDWY~~VEUX~~UDUY~~JEUP~~RNKY~~QNJY~~QLIY~~ZDRD~~MDGD~~ZY
RY~~NYGY~~YDWE~~XDWF~~TDUF~~SDUE~~KDJE~~HDJE~~WXYY~~WWXY~~UWTY~~UXSY~~KW
MY~~KWHY!!TDTY~~SESX~~RDRY~~WDOD~~WYHYHS~~UDTE~~UDTF~~QDRF~~PDRE~~TXUY~~
TWUY~~RWQY~~RXPY~~MYHX~~KYHW~~JYHU~~IYHS!!XDXX~~XDQY~~WDQU~~VDPU~~JDQY~~
JDJY~~IEIX~~HDHY~~EDUD~~JDED~~EYUY~~MYEY~~ZDXE~~GDHF~~FDHE~~XXZY~~XXUY~~
JXLY~~JWKY~~HWGY~~HXFY!!WDWX~~WDIY~~VDJU~~UDIU~~IEIY~~ZDUD~~LDFD~~ZYTY~~
YDWE~~KDIE~~GDIE~~WXYY~~WXUY!!QDTEUGWIXMXPWTUVTXQYOYOYLXJUITHPHMIIJGLEODQD
~~UGUIWLWQUTUV~~KUJTIQILJIKG~~QDSEUHULVQUUSXQY~~OYMXKUJQJLKHMEOD!!UDUY~~
UEUX~~TDTY~~YDMDJEIFHHHKIMJNMOTO~~JFIHIKJM~~MDKEJGJLKNMO~~YYQY~~XDUE~~WD
UF~~SDTF~~RDTE~~UXXY~~UWWY~~TWSY~~TXRY!!QDTEUGWIXMXPWTUVTXQYOYOYLXJUITHPHM
II JGLEODQD~~UGUIWLWQUTUV~~KUJTIQILJIKG~~QDSEUHULVQUUSXQY~~OYMXKUJQJLKHME
OD~~TUSTQSPSNTMUL\K↑I↑H\HZ~~LZK\JJIJ~~MUKCJ\I\HC!!UDUY~~UEUX~~TDTY~~YDMD
JEIFHHHJILJMMNTN~~JFIHIJJL~~MDKEJGJKKMMN~~PNNOMQKWJYHYGWGU~~KUJWIXHX~~NO
MPJUIWHWGU~~YYQY~~XDUE~~WDUF~~SDTF~~RDTE~~UXXY~~UWWY~~TWSY~~TXRY!!JGIDIJ
JGLEODRDUEWGWJULSNMPKQJSJUKX~~VJULSMMOKPJR~~UEUGUIUKSLMNJPIRIUJWKXNYQYTX
UUWSWYUU!!XDXJ~~QDQY~~PEPX~~ODOY~~HDHJ~~XDHD~~TYLY~~WDXJ~~VDXG~~UDXF~~SD
XE~~MDHE~~KDHF~~JDHG~~IDHJ~~QXSY~~QWRY~~OWNY~~OXMY!!WDWSUUTXQYOYLXJUISIE
~~UEUTUU~~UDUTTWSXQY~~ZDRD~~LDFD~~YDWE~~XDWF~~TDUF~~SDUE~~KDIE~~GDIE!!WD
PY~~UDPUPY~~UDOU~~IEPY~~YDRD~~MDGD~~XDUF~~TDUF~~SDUE~~KDIE~~HDIE!!XDTY~~
WDTTTY~~UDST~~PDSTTY~~PDLY~~ODLTLY~~NDKT~~HEKTLY~~EDSD~~PDND~~KDED~~ZDWE
~~YDWF~~UDUF~~TDUE~~JDHE~~FDHE!!WDKY~~UDJY~~UDIY~~JEUX~~YDRD~~MDGD~~YYSY
~~NYGY~~XDUF~~TDUF~~SDUE~~LDJE~~HDJE~~UXXY~~UXTY~~KXMY~~KWLY~~KWHY!!XDQO
```

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

TITLE	ABSTRACT NO
VARIABLE FONT PRESENTATION AID	51/00-8040/0

```
^^WKUL^^RKTL^^MKKL^^IKKL^^UXWY^^UXSY^^LXNY^^KXIY!!UKPY^^UKPW^^TKOW^^JLOW
RJT_U`X`Y_YJWEX_X^^^XKQK^^NKHK^^WKTM^^RKTL^^LKJL^^IKJL!!LKUY^^KKUY^^JK
TY^^JKUKUO^^UYJYJU^^UKUO^^TKUN^^SKUM^^QKUL^^OYJX^^MYJW^^LYJU^^KYJU!!
```

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

ROMAN ITALIC

0	-5	5
0	-5	6
88	-10	10
112	0	0
112	-10	11
215	0	0
216	-13	13
360	-5	5
384	-8	8
440	-8	8
510	-8	9
588	-12	13
620	-5	6
664	-12	13
680	-5	6
712	-11	12
728	-10	11
844	-10	11
884	-10	11
988	-10	11
1110	-10	11
1148	-10	11
1244	-10	11
1366	-10	11
1444	-10	11
1652	-10	11
1774	-5	6
1838	-5	6
1914	-5	6
1958	-12	13
1990	-5	6
2034	-10	11

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

[illegible]	0	0	[illegible]	0	0
[illegible]	-10	10	[illegible]	-11	11
[illegible]	-12	12	[illegible]	-9	10
[illegible]	-10	11	[illegible]	-9	9
[illegible]	-12	11	[illegible]	-11	11
[illegible]	-12	11	[illegible]	-9	9
[illegible]	-12	10	[illegible]	-8	8
[illegible]	-10	12	[illegible]	-10	11
[illegible]	-13	13	[illegible]	-11	11
[illegible]	-7	7	[illegible]	-7	6
[illegible]	-9	10	[illegible]	-7	6
[illegible]	-12	11	[illegible]	-11	11
[illegible]	-10	10	[illegible]	-6	6
[illegible]	-14	14	[illegible]	-18	17
[illegible]	-12	13	[illegible]	-12	12
[illegible]	-11	11	[illegible]	-10	10
[illegible]	-12	11	[illegible]	-11	11
[illegible]	-11	11	[illegible]	-11	10
[illegible]	-12	12	[illegible]	-9	9
[illegible]	-11	12	[illegible]	-8	9
[illegible]	-11	11	[illegible]	-7	7
[illegible]	-12	13	[illegible]	-12	12
[illegible]	-10	10	[illegible]	-10	10
[illegible]	-13	13	[illegible]	-15	15
[illegible]	-11	11	[illegible]	-11	11
[illegible]	-11	11	[illegible]	-11	11
[illegible]	-11	11	[illegible]	-10	10
51c0	0	0	[illegible]	0	0
51c0	0	0	[illegible]	0	0
51c0	0	0	[illegible]	0	0
51c0	0	0	[illegible]	0	0
51c0	0	0	[illegible]	0	0

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

```
H$
LDMDNEPR~~LEMEPR~~LELFPR~~LDKEKFPR~~RUSWSXRYQYPXPWQVRU~~RWRXQXQWRW!!RDSE
UK~~REUK~~RDQEUK~~HDIEKK~~HEKK~~HDGEKK!!N@VJ~~I@QJ~~HIHHIHIJGJGHHFIELDPD
SEUGUJTLRNLQKSKULX~~TJSLLPKR~~SETGTISKMNKPJRJUKWLXOYSYUXWWXUXSUSVUWUWT!!
FMFLGLGNENELFKGKILKNPURXTYWYZXEVETZRYQWPRNPMNKMIMGNEPDRESGSJRPQSPUNXLYJY
IWIU~~UYZX~~YXZUZTYRXQUP~~RNQQNWLX~~WYXXYUYTXRWQUPPM~~SJRMQPOTMWKXJXIW!!
MDNEPK~~MEPK~~MDLEPK!!H@JAMCPFRITMUQUUTZSJQ`~~OFQISMTRTZ~~H@KBNEPHQJRMSQ
TZ~~TRSER↑Q`!!O@MCLFKJKOLSNWPZSJU_X`~~LFLNMSOWQZ~~O@NBMELN~~LFMONSOUPXRE
U↑X`!!NDOEMONP~~NDNP~~SGRGJMIM~~SGIM~~SGSHILIM~~IGJGRMSM~~IGSM
~~IGIHSLSM!!PGPXOX~~PGOGOX~~XOGOGP~~XOXPGP!!QYRYSXSWRVQUPWPYQER\TJ~~RWRX
QXQWRW~~QYQZR\!!XOGOGP~~XOXPGP!!RUSWSXRYQYPXPWQVRU~~RWRXQXQWRW!!C@J`\`~~
C@B@\`!!NDQESGUJUMWQWTUWUXSYQYNXLVJSIPHLHIIFJELDND~~QFSHTJUMUQUUUW~~NWLU
KSJPILIHJF~~NDPERHSJTMUQUUTXSY~~QYOXMULSKPJLJGKELD!!NHSYQY~~KDMHRY~~KDQY
~~KDNGQISJ~~NHPISJ!!SISHRHRJTJTHSFREODLDIEHGHIIKKMUSWUYY~~JEIGIIJKLMOO~~
LDKEJGJIKKMMUS~~XWWUUUPWKWJU~~UUPXKX~~UUPYMYKXJUJU!!SISHRHRJTJTHSFREODLD
IEHGHIIKJLLMON~~JEIGIIJKKL~~LDKEJGJIKKMMON~~QNONLOKPJRJUKWMXPYSYUXWWXUXS
VSUUWUWT~~LPKRKULW~~ONMOLQLUMWNXPY!!KHPYNY~~HDJHOY~~HDNY~~HDXSHS!!QDUN~~
QDGD~~QEIE~~RFMFIEGD~~VNUMRLOLLMKNJPJSKVMXQYTYUXWWXUXSUSVUWUWT~~LNKPKSLV
HX~~OLMMLOLSMVOXQY!!IHIGJGJIHIHGIEKDNDQESGUJUMWQWTUWUXSYPYMXKVJTJQKOLNNM
QMSNTOUQ~~RGTJUMVQUUUW~~LUKTKQLO~~NDPERHSJTMUQUUTXSY~~PYNXMWLTLPMNNM!!TD
VJ~~GDHGJJNOPRQURY~~PQRUSY~~JJPPRSSUTYRY~~UGRDPDKG~~SEPEKG~~UGSFPFKGIGHF
GD!!ODRESFTHTKSMQNNNKMILHJHGIEKDOD~~MDRE~~RFSHSLRM~~SMPN~~ONKM~~JLIJIGJE
~~IEMD~~ODQFRHRLQN~~NNLMKLJJJFKD~~QNUOWQXSXUWXTYPYLXKWJUJRKPLONN~~PNUO~~
TOVQWSWVUX~~WXRYLX~~LWKUKRLP~~LOON~~QNSOUQVSVUUXTY~~PYNXMWLULQMONN!!JLKN
LONPQPSOTNULUITGREODLDJEIFHIHLIPJSLVNXQYTYUXWUWTUTUVUVUU~~SNTLTISG~~JFIH
ILJPKSMU~~QPROSMSIRFQEOD~~LDKEJGJLKPLSMUOXQY!!OKPLPMONNNMMMLNKOK~~OLOMNM
NLOL~~RVSWSXRYQYPXPWQVRU~~RWRXQXQWRW!!OKPLPMONNNMMMLNKOK~~OLOMNMNLOL~~QY
RYSXSWRVQUPWPYQER\TJ~~RWRXQXQWRW~~QYQ2R\!!KDMENFOHOJNKMKLJLIMHNH~~MENGNH
~~NINJMJMINI!!XKGKGL~~XKXLGL~~XSGSGT~~XSXTGT!!LGMGNFNEMDLDKEKGLIMJOK~~ME
MFLFLEME~~LGLHMJ!!SISHRHRJTJTHSFREODKDHEGGGIHKILKMONQQQQRNR~~MDHE~~IEHG
HIIKJLLM~~KDJEIGIIJKKLONPOPQQR~~RVSWSXRYQYPXPWQVRU~~RWRXQXQWRW!!MDYX~~OH
NY~~NFMX~~MDMFLWLY~~USNS~~\YUY~~QYJY~~YXCY~~YXWY~~NXPY~~NWOY~~LWKY!!SDYY
~~RDXY~~QDWY~~UDKDHEGGGIHLIMLN~~IEHGHIILJM~~KDJEIGIIJLLN~~TNLNJOIQISJULX
PY\Y~~KOJQJSKUMX~~LNKPKSLUNXPY~~UDRE~~TDSF~~PDRF~~ODRE~~XXCY~~XWZY~~WWUY
~~XXUY!!HFGFFDGJGHHFIEKDNDQESGUJUMWQWTUWUXRYOYMXKVJT~~QFSHTJUMVQVUUUW~~ND
```

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO

51/00-8040/0

```
PERHSJTMUQUUTXRY!!SDYY~~RDXY~~QDWY~~UDMDJEIFHIHMIQKUMWOXSY\Y~~KEJFIIIMJQ
LUNW~~MDKFJIJMKQMUPXSY~~UDRE~~TDSF~~PDRF~~ODRE~~XXEY~~XWZY~~WWUY~~XXUY!!
SDYY~~RDXY~~QDWY~~MJOR~~UDGDHJ~~TNNN~~\YMYKT~~UDRE~~TDSF~~PDRF~~ODRE~~KD
HE~~JDHF~~IDHG~~HDHJ~~MJONOR~~NLPNOP~~NMQNOO~~XXEY~~WWUY~~XXUY~~RY
MX~~PYMW~~MWKT!!SDYY~~RDXY~~QDWY~~MJOR~~UDGDHJ~~TNNN~~\YTY~~UDRE~~TDSF~~
PDRF~~ODRE~~KDHE~~JDHF~~IDHG~~HDHJ~~MJONOR~~NLPNOP~~NMQNOO~~XXEY~~XWZY~~
WWUY~~XXUY!!HFGFFDGJGHHFIEKDNDQESGUJUMWQWTUWUXRYPYMXKUIR~~QFSHTJUMUQUUUW
~~LUKUJR~~NDPERHSJTMUQUUTXRY~~PYNXLUKR~~NRFR~~MRKS~~LRKU~~HRJT~~GRJS!!TD
ZY~~SDYY~~RDXY~~HDNY~~GDMY~~FDLY~~WDOD~~KDCD~~UNJN~~JYUY~~QYIY~~UDSE~~UD
TF~~QDSF~~PDSE~~JDGE~~IDHF~~EDGF~~DDGE~~YX\Y~~YWEY~~XWWY~~YXUY~~MXPY~~MW
OY~~LWKY~~MXJY!!NDTY~~MDSY~~LDRY~~QDID~~WYOY~~PDME~~ODNF~~KDMF~~JDME~~SX
UY~~SWUY~~RWQY~~SXPY!!KDPUQWSY~~JDNQOTPU~~IDMQOUQXSYUYWXXUXTWSUSUTUUUWU
~~WTWUUUTWT~~NDFD~~MDJE~~LDKF~~HDJF~~GDJE!!SDYY~~RDXY~~QDWY~~FEUP~~QMMY
~~PWLY~~OLKX~~UDND~~IDCD~~\YTY~~PYIY~~UDRE~~TDSF~~PDRF~~ODRE~~HDFE~~DDFE
~~XWZY~~WWUY~~XXUY!!ODRETGUJWMXQXTWWUXTYQYNXLUJSIPHLHIIFJELDOD~~SGUJUMWQ
~~XXEY~~XWZY~~WWUY~~XXUY~~MXOY~~MWNY~~LWJY!!QDWY~~PDUY~~ODUY~~TDLD~~ZYKY
IS~~SDPE~~RDQF~~NDPF~~MDPE~~UXYY~~UWXY~~UWTY~~UXSY~~PYKX~~NYJU~~LYIS!!UD
EX~~UETWTY~~TDSW~~SDRU~~GDRUTY~~GDMY~~FDLY~~EDKY~~XDSD~~GDBD~~YXY~~PYHY
~~WDUE~~UDUF~~DDFF~~CDFE~~EXJY~~EXYY~~LXOY~~LWNY~~KWJY~~LXIY!!SDYX~~SDLY
~~RDLU~~QDKU~~FEKULY~~UDQD~~IDCD~~\YUY~~UDRE~~TDRF~~HDFE~~DDFE~~YXEY~~YX
WY!!ODRETGUJWMXQXTWWUXTYQYNXLUJSIPHLHIIFJELDOD~~SGUJUMWQWUUW~~MUKSJPILIH
JF~~ODQESHTJUMVQUUUXTY~~QYOXMULSKPJLJGKELD!!SDYY~~RDXY~~QDWY~~UDJDGEFGFI
GLINMOUO~~HEGGGIHLJN~~JDIEHGHIILKNMO~~\YTY~~UDRE~~TDSF~~PDRF~~ODRE~~XXEY
~~XWZY~~WWUY~~XXUY!!ODRETGUJWMXQXTWWUXTYQYNXLUJSIPHLHIIFJELDOD~~SGUJUMWQ
WUUW~~MUKSJPILIHJF~~ODQESHTJUMVQUUUXTY~~QYOXMULSKPJLJGKELD~~UUUTSSRSPTOU
NEM\L\KE~~N\MILJ~~OUOJN↑L↑KEKZ!!SDYY~~RDXY~~QDWY~~UDKDHEGGGIHLIMLNTN~~IE
HGHIILJM~~KDJEIGIIJLLN~~PNNOMPKUJWIWHU~~KWJXIX~~MPLXKYIYHVHU~~\YTY~~UDRE
~~TDSF~~PDRF~~ODRE~~XXEY~~XWZY~~WWUY~~XXUY!!HFGFFDGJGHHFIELDPDSEUGUJTLRN
LQKSKULX~~TJSLLPKR~~SETGTISKMNKPJRJUKWLXOYSYUXWWXUXSYYXWWW!!NDTY~~MDSY~~
LDRY~~UDWJ~~EDFJ~~UDED~~WYOY~~TDWJ~~RDVG~~PDUE~~IDFE~~HDFF~~GDFG~~FDFJ~~
SXUY~~SWUY~~RWQY~~SXPY!!TDWOXSXUWXTYPYMXKUJSFE~~SDUOWSWWUX~~RDUOUSUWTY~~
WDOD~~IDCD~~UDSE~~UDTF~~QDSF~~PDSE~~HDFE~~DDFE!!TDTFSWSY~~SERU~~RDQU~~GE
SY~~UDOD~~JDDD~~UDTF~~QDRF~~PDSE~~IDGE~~EDGE!!UDUFWWWY~~TEUU~~SDUU~~MDUU
WY~~MDMFOWOY~~LENU~~KDMU~~EEMUOY~~XDPD~~MDKD~~HDBD~~WDTE~~\DUF~~RDTG~~QD
TE~~GDEE~~CDEE!!TDNY~~SDMY~~RDLY~~GEYX~~UDOD~~JDDD~~\YUY~~QYJY~~UDSF~~QD
RF~~PDRE~~IDGE~~EDGE~~YXEY~~YXWY~~NXPY~~NWOY~~MWKY!!UDQNTY~~TDPNSY~~SDON
```

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

```
RY~~FEON~~WDPD~~IDCD~~WYOY~~VDTE~~RDSF~~QDTE~~HDFE~~DDFE~~SXUY~~SWUY~~RW
QY~~SXPY!!HD2Y~~GDYY~~FDXY~~FDTDUJ~~ZYLYJS~~SDUJ~~RDUG~~PDTE~~PYLX~~NYKU
~~NYJS!!KKMRMULXKYIYGWFU~~JKLRLX~~KKIKKRLU~~MRMONLPKRKULWOXRXTWWUXTYRYPX
OWNUMR~~TLUOWRWUUW~~RKTMUOURVUUXTY!!RDTKUQUUTWSXQYOYLXJUIRIPJMKLMKOKQLRM
SOTR~~QDSKTOTUSX~~LWKUJRJOKM~~UDPDRKTR~~OYMWLUKRKOLLMK~~TDQE~~SDRF!!KOKN
LNLPJPJNKLMKPKSLUOVRVTUWTXRYPYMXKU~~SMTOURUUTW~~PKRMSOTRTUSXRY!!IDLOMSMU
LXKYIYGWFU~~HDKOLSLX~~LDGDKRLU~~MRMONLPKRKULWOXRXTWWVXTYRYPXOWNUMR~~UMVO
WRWUUW~~RKTMUOVRUUUXTY~~KDHE~~JDIF!!UTQSNRKPJNKLMKPKSLUOURUTUWTXRYPYMXKU
~~SMTOURUUTW~~PKRMSOTRTUSXRY!!HFHEIEIGGGGEHDJDLENGOIPLQPSYT\U↑W\~~WHOKPP
RYS\~~JDLFMHNKOPQXRESJU_W\Y\Z_ZJXEJX_Y_Y↑~~TKIK!!JKNYO\Q_S\~~IKMYOJ~~JKHK
LYWJP_S\U\X_Y↑Y\W\W↑X↑XJ~~LRLOMLOKQKTLVOWRWTVWUXSYQYOXNWMULR~~TMUOVRVUUW
~~QKSMTOURUUTXSY!!SDYYWY~~RDXY~~VDQDWY~~VDQDWY~~IMILJL
MSMULXKYIYGWFU~~UDRE~~TDSF!!ODOFMFMDOD~~NDNF~~OEME~~VOUMSKQKPLONQQQU~~PL
PPQTQX~~PNRSRVQXPYNYLWKU!!MDMFKFKDMD~~LDLF~~MEKE~~UOTMRKPKOLNNNQPXQERJT_
U\X\Y_YJWJW_X_X↑~~OLOQQXRESJ~~ONPRRYS\T↑V\!!SDYYWY~~RDXY~~VDQDWY~~IMILJL
JNHNWLIKKKMLQPSQ~~UQSQQRPSNWMXKX~~QSOWNX~~SQRRPXOYMYKXIU~~UDRE~~TDSF!!ND
QORSRVQXPYNYLWKU~~MDPOQSQX~~QDLDPRQU~~PDME~~ODNF!!QO`M↑K\KELZNZQ\Y~~ELEQ
JY~~EN\R↑Y\Y~~ZQXNULTKRKPLONQQQY~~PLPQRY~~PNQRSYQY~~OQMNKLIKGKELDNDQFU~~
ELEPFTFX~~ENGSGVFXEYCYAWQU!!EOZMXKUKULTNTQUV~~ULUQWY~~UNVRXYUY~~TQRNPLNK
LKJLINIQKU~~JLJPKTKX~~JNLSLVKXJYHYFWEU!!QKTLVOWRWTVWUXRYOYLXJUIRIPJMKLNK
QK~~TMUOVRVUUW~~LWKUJRJOKM~~QKSMTOURUUTXRY~~OYMWLUKRKOLLNK!!ZOYMWKUKTLSN
SQTUW\~~TLTQUUX\~~TNURY\~~SRRQQMPLNKLKJLIMHPHRIUKXNYPYRXSUSR~~JMIOIRJUKW
~~LKKLJOJRKULWNY~~\T\~~X_E\~~X↑Z\~~W↑U\~~X_U\!!KKQ\~~JKP\~~KKIKO\~~MRMO
NLPKRKULWOXRXTWWXTYRYPXOWNUMR~~UMVOWRWUVW~~RKTMUOVRVUUXTY~~T\L\~~P_S\~~
P↑R\~~O↑N\~~P_M\!!XOWMUKSKRLQNQRSY~~RLRRTY~~RNSRUYSY~~IMILJLJNHNHLIKKKML
ONQR!!JHJMKMKOIOIMJLMKPKSLTMTOSQQRNSLTKU~~SLTO~~SPQQNRLS~~KTLX~~TMSOQPNQ
LRKTKULXOYRYUXUWUUTUTWUWUU!!NDQQRSRVQXPYNYLWKU~~MDPOQSQX~~NDLDPRQU~~TKJK
!!EOZMXKUKULTNTQVU~~ULUPUTUX~~UNWSWUVXTYRYPXNULS~~JKLSLUKXJYHYFWEU~~IKKS
KX~~JKHKJRKU!!YOXMUKTKSLRNRQTU~~SLSPTTTX~~SNUSUUTXRYPYNXLUJSIOIKJKJLIN!!
↑OJMEKYKXLWNWQYU~~XLXPYTYX~~XNZSZUYXWYUYSXQVPS~~NKPSPVOXMYKYIXGVESDODKEK
ELDN~~MKOSOX~~NKLKNROV!!XOULTKRKPLONOP~~RKQLQPRTSVUXWYYYZXZVXVXXYXYW~~PM
PPQTQW~~HMHLILINGNGLHKJKLLNNOPPTPXOY~~RTRVQXOYMYKXIU!!ZOYMWKUKTLSNSQUV~~
TLTPUTUX~~TNVSVUVXSYQYOXMVKR~~IKMYN\P_R\~~HKLYNJ~~IKGKKYMJO_R\U\W_X↑X\V\
V↑W↑WJ!!IKJMLOTUVWWY~~JMSMUNVP~~LMPLSLTM~~LMPKSKUMVP~~VWMWKVJT~~TWPXMXLW
~~TWPYMYKWJT!!
```

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-8040/0

	0	-9	9
GOTHIC	0	-5	6
	-8	-9	9
	9<	0	0
	9<	-9	10
	194	0	0
	194	-1<	13
	335	-4	5
	358	-7	7
	-1[illegible]	-[illegible]	[illegible]
	-54	-8	8
	53<	-1<	13
	554	-5	6
	61<	-1<	13
	628	-5	6
	661	-11	1<
	67[illegible]	-1[illegible]	10
	761	-10	10
	734	-10	10
	85<	-10	10
	948	-10	10
	984	-10	10
	1091	-10	10
	1<14	-10	10
	1<38	-10	10
	1374	-10	10
	1498	-5	6
	156<	-5	6
	164<	-5	6
	1691	-1<	13
	17<<	-5	6
	1770	-9	10

TITLE	ABSTRACT NO:
VARIABLE FONT PRESENTATION AID	51/00-8040/0

1886	0	0	3c80	0	0
1886	-10	10	3c80	-10	10
19c6	-10	10	335c	-10	10
c014	-10	11	34c4	-9	9
c090	-10	11	3488	-10	10
c154	-9	10	3560	-9	9
c208	-9	9	363c	-6	8
c250	-10	11	3680	-10	10
c338	-11	11	3776	-10	10
c38c	-8	9	38c8	-4	5
c428	-8	9	38f4	-4	5
c468	-10	11	39cc	-10	9
c51c	-9	8	3966	-4	5
c540	-1c	1c	398c	-15	16
c59c	-11	11	4066	-10	10
c63c	-11	11	4116	-9	10
c71c	-10	10	4188	-10	10
c76c	-11	11	4260	-10	10
c86c	-10	10	433c	-7	7
c930	-10	10	4374	-8	9
3016	-8	9	4474	-5	6
3046	-11	11	4506	-10	10
3094	-10	10	4554	-8	8
31cc	-13	13	4584	-1c	1c
3174	-10	10	4636	-9	9
3c06	-9	10	4683	-8	8
3c40	-10	10	470c	-9	9
3c80	0	0	474c	0	0
3c80	0	0	474c	0	0
3c80	0	0	474c	0	0
3c80	0	0	474c	0	0
3c80	0	0	474c	0	0

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

```
H$
PDPROR~~PDODOR~~PUQWQXPYOYNXNWOUPU~~PWPXOXOWPW!!TDUEUK~~TEUK~~TDSEUK~~KD
LELK~~KELK~~KDJELK!!PePJOJ~~PeQeOJ~~KGIGKENDQDTEVGUIUKTLLPKQJSJUKWNXQXSW
TU~~KGLFNEQETFUGUITKLOJQISIUJWKXNYQYTXVUTU~~JUMX!!FKHKJLKNMTNUOWQXUXWWXU
XSWQVPQMOKNINGOEQDRDTEUGUITLROMUJXHYFY~~FKFLHLJM~~ILJNLTMUOXQYUYWXXWYUYS
XQVOQLPKOIOGPE~~OFQERETF~~SETGTISLQOLUJWHXFXFY!!ODPEPK~~OEPK~~ODNEPK!!Me
OBQESITNTRSWQEO↑M`L`~~MeL@NBPERISNSRRWPEN↑L`!!T@RBPENIMNRNWPER↑T`S`~~T@
S@QBOEMILNLRMWOEQ↑S`!!PDQEOOPP~~PDPP~~PDOEQOPP~~UGTGLMKM~~UGKM~~UGUHKLKM
~~KGLGTMUM~~KGUM~~KGKHULUM!!PGPXOX~~PGOGOX~~XOGOGP~~XOXPGP!!PUQWQXPYOYNXNWOUPU~~
OVNWNZO\QJ~~PWPXOXOWPW~~OYN2~~NXO\!!XOGOGP~~XOXPGP!!PUQWQXPYOYNXNWOUPU~~
PWPXOXOWPW!!G@Y`X`~~G@F@X`!!QDTEVHWMWPVUTXQYOYLXJUIPIMJHLEODQD~~SEUHVMUP
UUSX~~TWQXOXLW~~MXKUJPJMKHME~~LFOEQETF!!THRGODOY~~THTIRHPFPYOY!!VIVHUFTE
RDNDLEKFJHJJKLMOUY~~VIUIUHTFRENELFKHKJLLNOWY~~VXIXIY~~WYIY!!UDJDQM~~UDUE
KE~~KDRM~~QLOLLMJOIRISJULXOYRYUXUWWUVU~~RMOMLNJQ~~NMKOJRJSKVNX~~JTLWOXRXU
UWVU~~SXUV!!MGMYLY~~LDLY~~LDWTHT~~MGVT~~USHSHT!!UDVM~~TEUL~~UDKDKE~~TEKE
~~ULRKOKLLJNIQISJULXOYRYUXUWWUVU~~VMUMSLOLLMJP~~NLKNJQJSKVNX~~JTLWOXRXUW
VU~~SXUV!!LEKGJGKENDPDSEUHVMVRUVSXPYOYLXJUISIRJOLMOLPLSMUO~~KFNEPESF~~RE
THUMURTVQX~~UTSWPXOXLWJT~~NXKVJSJRKONM~~JQLNOMPMSNUQ~~QMTOUR!!WDIDSY~~WD
WEJE~~JDTYSY!!RDUEUGVIUKTLRMNNLOKPJRJUKWNXRXUWVUVRUPTORNNMLLKKJIJGKENDRD
~~TEUGUITKRLNMLNJPIRIUJWKXNYRYUXUWWUWRVPTNRMNLLKKIKGLE~~KFNEREUF~~VVSX~~
MXJU!!KNMPPQQQTPUNWKWJVGTEQDPDMEKGJKJPKUMXPYRYUXUVUUVTX~~KKLNOP~~KLMOPPQP
TOUL~~RPUNVKVJUGRE~~VITFQEPEMFKI~~OELGKKKPLUNX~~MWPXRXUW!!PKQLQMPNONNMNL
OKPK~~PLPMOMOLPL~~PVQWQXPYOYNXNWOVPU~~PWPXOXOWPW!!PKQLQMPNONNMNLOKPK~~PL
PMOMOLPL~~NXOYPYQXQWPVOUNWNZO\QJ~~PWPXOXOWPW~~OYNZ~~NXO\!!NDPEQGQJPKOKNJ
NIOHPHQI~~PIPJOJOIPI~~PEQI~~QGPH!!XKGKGL~~XKXLGL~~XSGSGT~~XSXTGT!!NFOGPG
QFQEPDODNENHOJQK~~PEPFOFOEPE~~OGNH~~NFOJ!!VIVHUFTEQDNDKEJFIHIJJLKMMNPO~~
VIUIUHTFQENEKFJHJJKLMMPN~~UGRE~~MEJG~~JKNN~~PNPRORON~~PVQWQXPYOYNXNWOVPV
~~PWPXOXOWPW!!PDXY~~PGWYXY~~PGIYHY~~PDHY~~USKS~~UTJT!!VDVY~~UEUX~~VDNDKE
JFIHIKJMKNNO~~UENEKFJHJKKMNN~~UNNNKOJPIRIUJWKXNYVY~~UONOKPJRJUKWNXUX!!HI
IGKEMDQDSEUGVIWLWQVTUVSXQYMYKXIVHT~~HIIIJGKFMEQESFUIVLVQUTSWQXMXKWJVITHT
!!VDVY~~UEUX~~VDODLEJGIIHLHQITJVLXOYVY~~UEOELFKGJIILIQJTKVLWOXVX!!VDVY~~
TETX~~UDID~~TEIEID~~TNNNNO~~TONO~~TXIXIY~~VYIY!!VDVY~~TETYVY~~VDID~~TEIE
ID~~TNNNNO~~TONO!!HIIGKEMDQDSEUGVIWLWQVTUVSXQYMYKXIVHTHPMP~~HIIIJGKFMEQE
SFTGVIVLVQVTTVSWQXMXKWJVITIQMQMP!!WDWY~~WDVDVYWY~~IDJDJYIY~~IDIY~~VNJN~~
VOJO!!PEPY~~OEOYPY~~VDIDIE~~VDVEIE~~VYIYIX~~VYVXIX!!LDLTMWOXQXSWTTUT~~LD
```

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

SCRIPT

0	-8	8
0	-5	6
34	-9	9
58	0	0
58	-10	11
140	0	0
140	-13	13
[illegible]	-4	5
[illegible]	-7	8
[illegible]	-8	7
[illegible]	-8	9
[illegible]	-13	13
[illegible]	-5	6
[illegible]	-13	13
[illegible]	-5	6
[illegible]	-11	11
[illegible]	-10	11
496	-10	11
566	-10	11
610	-10	11
710	-10	11
730	-10	11
808	-10	11
900	-10	11
960	-10	11
1086	-10	11
1178	-5	6
1200	-5	6
1228	-5	6
1244	-13	13
1256	-5	6
127[illegible]	-10	11

TITLE	ABSTRACT NO.
VARIABLE FONT PRESENTATION AID	51/00-8040/0

1340	0	0	368c	0	0
1340	-13	10	368c	-7	9
1416	-1c	1c	3748	-6	8
1556	-10	11	3794	-6	6
1638	-1c	11	3838	-7	9
17c4	-9	10	3904	-6	6
18cc	-11	10	3950	-3	6
19[illegible]	-11	11	4[illegible]	-7	9
c0c0	-1c	1c	4080	-6	9
c130	-9	7	4140	-4	4
c196	-9	8	4178	-4	4
c2b4	-1c	1c	4226	-6	8
c382	-9	9	4288	-4	4
c46c	-14	14	43cc	-13	1c
c50[illegible]	-11	1c	44[illegible]c	-9	9
c636	-10	11	4476	-7	7
c7[illegible]4	-1c	11	4534	-6	9
c810	-10	11	4594	-7	9
c896	-1c	1c	4664	-6	8
30c6	-10	10	47[illegible]c	-4	8
3098	-9	9	4756	-4	4
31[illegible]4	-11	11	480c	-7	9
3c5c	-11	10	4856	-7	8
33c4	-1c	11	490c	-10	11
3406	-10	10	49[illegible]4	-8	8
3510	-11	11	5[illegible]8	-7	9
3594	-11	10	51cc	-6	7
368c	0	0	5198	0	0
368c	0	0	5198	0	0
368c	0	0	5198	0	0
368c	0	0	5198	0	0
368c	0	0	5198	0	0

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO.

51/00-8040/0

```
H$
MDNEPQ~~MEPQ~~MDLEPQ~~RWSXRYQXRW!!RDTK~~QDTK~~IDKK~~HDKK!!N@VJ~~1@@J~~HH
 IIHJGIGHHF1ELDPDSEUGUITKSLLPJR~~UISKLOKPJRJUKWLXOYSYUXWWXUXTWSUTWU!!FLGM
FNEMELFKGKILKNPURXTYWYZXCVCT2RYQWPRNPMNKMIMGNEPDRESGSJRPQSOUMXKYIYHWHU~~
WYYXZUZTYRXQRN~~SJROQROUMWKXIXHW!!MDOK~~LDOK!!H@LCOFQISMTRTVSCR↑Q`~~LCOG
QKRNSSSXRJQ`!!O@NBMELJLNMSOWQZTJX`~~O@NCMHMMNROUQYTJ!!NDNP~~SGIM~~IGSM!!
QYNXLVJSIPHLHIIFJELDND~~NDPERGTJUMVQVTUWSY~~QYOXMVKSJPILIIJFLD!!NHSY~~LD
RY~~LDOGRITJ~~MGQITJ!!SHRISJTITHSFREODLDIEHGHIIKKMNORQUSWUYY~~LDJEIGIIJK
LMRQ~~XWWVUVPXMXKWJU~~UVPYMYKXJU!!SHRISJTITHSFREODLDIEHGHIIKLMON~~LDJEIG
II JKLM~~QNONLOKPJRJUKWLXOYSYUXWWXUXTWSVTWU~~ONMOLPKRKULWMXOY!!JEPY~~IDOY
~~IDXSHS!!QDVN~~QDGD~~QELEGD~~VNUMRLOLLMKNJPJSKVMXPYSYUXWWXUXTWSVTWU~~OL
MMLNKPKSLVNXPY!!IGJHIIHHHGIEKDNDQESGUJVMWQWUVWUXSYPYMXKUJTJQKOLNNMQMSNUP
VR~~NDPERGTJUMVQVUVUX~~PYNXLVKTKPLN!!TDVJ~~GDHGJJOPQSRUSY~~JJPPRSSUTY~~UG
RDPDKG~~TFREPEKGIGHFGD!!ODRESFTHTKSMQNNNJMILHJHGIELDOD~~ODQERFSHSKRMQN~~
NNKMJLIJIGJELD~~QNUOWQXSXUWXTYPYLXKWJUJRKPLONN~~QNTOVQWSWVUXTY~~PYMXLWKU
KQLO!!IKJMLONPQPSOTNULUITGREODLDJEIFHHHLIPJSLVNXQYTYVXWVWUVTUUVU~~SOTMTI
SGQEOD~~JEIGILJPKSMVOXQY!!OKPLOMNLOK~~RWSXRYQX!!OKPLOMNLOK~~RYSXRWQXQYRE
HKILONQQQQPRNR~~KDIEHGHIIKJLLM~~RWSXRYQXRW!!JDLFNIQNSQVUYXEYJY↑X↑VJEU\VJEW
~~JDKHMRNY~~JDMY~~NYNWOTPRRPTOVOWPWRVUUSXPYLYJX!!MENFOHQMSSTUVXXY~~NFOIQQ
RTSVUXXYZYEXEVZUYUZW~~SJTLUMWMXLXJWHUFSEPDJDHEGGGIHKJLNMPM~~JDIEHGHIIKJL
~~NMKNJOIQITJWKXMYOYPXPVOS~~NMLNKOJQJTKWMY!!WFXHXJWLTMQMMLKKIIHGHEIDKDNE
QHSKUOVSVUUXRYPYMXKVJTJRKPMPOQPS~~KDMEPHRKTOUSUUTXRY!!MENFOHQMSSTUVXXY~~
NFOIQQRTSVUXXYZYEXEVZUXUVUTXRYOYMXKVIRHMHJIGKEMDRDUEWGXIXKWLULTKSI!!KGLH
LJKKIKHIHGIEKDNDPEQFRHRJQLOM~~NDPFQHQKOM~~OMQMTNVPWRWUVWUXSYPYMXKVJTJRKP
MPOQPS~~QMSNUPURVUUX!!KFLHNMPSQUSXUY~~QJRLTMVMWKWIVGTEQDGDJEKFLINQOTPVRX
UYHYYXZWZUYUXUYW~~ODKEJE~~SQRPPOLOJNHKJR!!XGYIYKX!!NSNPMNLKIJFJEKDLDNEPG
QIRLRQQQORMRKQIOHM~~KDMEOGPIQLQPOR~~HMIQKUMWOXSYUYXXYUYUXTWUXU~~IQKTMVPX
SY!!VJWIWGVESDPDSOUUVWWXYYCY\X\VEUZUEW~~PDSMTPVUWWYY~~XRWQUPLMJLGJEHDFDE
EDFDHEJHKJMPNTNWLYKYIXGV~~FDHFJJLPMTMWLY!!KFMIONQSRUTXUY~~IJKLNMQMSLTJTH
SFQEMDIDKFLHNNPTQVSXUYXYYXYUXUWUXW!!IDKFMIONRWTE~~IKKMNNQNSMTKTISGQEMDID
KGLIORQVRXTEU\WJX\XZWXUUSUPTLS!!VJWIWGUERDPDSOUUVWWXYYCY\X\VEUZUEW~~PDSM
TPUUWWYY~~HEKIMKOLRM~~EEFFEGDFDEEDFDHEKJLKNLRM~~RMONNPMWLY~~RMPNQPNWLYKY
IXGV!!UGVIVKUMSHPNMMKLHIGFGEHDIDKELFNIRSSUUXWY~~LFNJPQQTRVTXWYYYZXZVYUWU
```

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

```
UURXPYMYKXIV!!PDTMWSYUEXJY_Y`X`U_U↑U_W~~PDRKSOTTTXRY~~PDQHRMSTSXRY~~GDKM
PURY~~GDIKJOKTKXIYHYFXDU~~GDHHIMJTJXIY!!PDQHSNUSVUXXZY\YJX\JU\UEU\W~~PDPI
OTNY~~PDOINTNY~~BECFBGAFAEBDDDFEHHIJKOMUNY!!ODQESGUJULWPWTVWUXSYQYNXLVJS
IQHMHIIFJEKEMFOHQLRQRT~~QESHULVPUTUWSY!!MENFOHQMSSTUVXXY~~NFOIQQRTSUUXXY
ZYCXEVZUYUZW~~SJTLUMWMXLXJWHUFSEPDLDIEHFGHGKHMINLONOPH~~LDJEIFHHHKIMJNLO
!!MHMJNLOMQNSNTLTJSGQENDKDIEHGHKINKQOURWTXWYYYZXZUYUWUUVRXOYLYJXHU~~KDJE
IGIKJNLQOTSWWY!!MENFOHQMSSTUVXXY~~NFOIQQRTSUUXXYZYCXEVZUYUZW~~SJTLUMWMXL
XJWHUFSEPDKDHEGGGIHKILLMPM~~KDIEHGHIIKJLLM~~PMMNLPKWJY~~PMNNMPLWJYIYGXEV
!!TGUIUKTMRNONLMJLGIFFEGDHDJEKFLHMKORPURXTY~~LHMLNSOUQXTYWYYXZVZUYTXUYV
!!IFJHLMNSOUQXSY~~OJPLRMTMUKUITGREODFDHEIFJILQMTNVPXSYUYWXXWXVWUUVWW~~MD
IEHE!!ZHXEVDUDSFSITLWTWWUY~~UDTFTIWQXTXWUYTYRXOUMRLP~~HDLPMTMWKYJYHXFU~~
GDKPLTLWKY!!ZHXEVDUDSFSITMVTVWUY~~UDTFTIVPWTWUYTYQXNULRJNIKHGHEIDJDKELG
LJKLINGOEO!!YJZJCIEGZEXDTDUFUJWSXY~~VJVSWY~~LDNFPJSSUWWY~~LDMFNJOSPY~~NJ
NSOY~~BDDEFGHJKSMWOY!!RISJUJVIVGUESD@DOENGNJONQSSVUXXYZYCXEVZUYUZW~~QDPE
OGOJPNRSTUVXXY~~EEFFEGDFDEEDGDIEKGMJONPSPUOXNYMYKXIV!!XHVETDSDQEQGSMSPRR
~~SDRERGTMTPRRPRMQKOILHJ~~FDHJKRMV~~EDGJIOKSMVOXRYUYXXYUYUXTWUXV!!HFIHKM
LPMROUQWSXUY~~OJPLRMTMUKUITGREODEDGEHFIIJMLSNVQXUYZYCXEVZUXUVUSXQYNYKXIV
~~LDHEGE!!MSNQPPRPTQURVTUVUUXSYQYOXNV~~RPTRUTUWSY~~LPNVNXLYJXIWGT~~KPMVMX
LY!!UTTQRM~~ODUVUXSYRYPXNVMSMPLTKUJUHT~~NDTVTXSY!!NQORNRNQOPQPSQTRUTUVTX
RYOYLWJT~~QPSRTTTWRY!!MSNQPPRPTQURVTUVUUXSYQYOXNV~~RPTRUTUWSY~~HDNVNXLYJX
IWGT~~GDMVMXLY!!SWQUPUOSOQPPQPSQTRUTUVTXRYOYLWJT~~QPSRTTTWRY!!PPMMKJJGJE
KDMENGWbWdVeTdSaRXQYOYMXLWJT~~NGOLPPSYU↑Wb!!MSNQPPRPTQURVTUVUXSYQYOXNV~~
RPTRUTUWSY~~LPRb~~KPNYP↑RbSdUeVdVbU_SJPELYIWGT!!UTTQRM~~ODUY~~NDUY~~SSQQ
OPNPLQLSMVMXLY~~NPMQMSNVNXLYJXIWGT!!OJPKOLNKOJ~~QPSVSXQYOXNWLT~~PPRVRXQY
!!OJPKOLNKOJ~~QPWb~~PPSYU↑WbXdZeEdCbZ_XJUEQYNWLT!!UTTQRM~~ODUY~~NDUY~~MP
MQLQMPNPPRSS~~SSPTOXNY~~SSQTPXNYMYJWHT!!TTRQPM~~MDSVSXQYOXNWLT~~LDRVRXQY
!!JTEQYPWQWSYY~~YPXQXSZY~~WSUQSPRPPQPSRY~~RPQQQSSY~~PSNQLPKPIQISJVJXIY~~
KPJQJSKVKXIYGXFWDT!!YTWQUPSQSSUY~~UPTQTSUY~~SSQQOPNPLQLSMVMXLY~~NPMQMSNV
NXLYJXIWGT!!PPRPTQURVTUVUXSYQYOXNWMUMSNQPPQQQSPUNVLVJUIT~~RPTRUTUWSY!!VT
TQRM~~QJZe~~PJYe~~SSQQOPHPLQLSMVMXLY~~NPMQMSNVNXLYJXIWGT!!MSNQPPRPTQURVT
UUUXSYQYOX~~RPTRUTUWSY~~LPRbRdQeOdNaNYLYIWGT~~KPNYP↑Rb!!VTTQRPPQPSRY~~RP
QQQSSY~~PSNQLPKPLS~~LPLSKUJUHT!!TTRQQQQNSMUMWNXPY~~QQOSNUNWPY~~TXRYMYJW
HT!!TTRQPM~~MDSVSXQYOXNWLT~~LDRVRXQY~~TLJL~~TKJK!!TPVVVXTYSYQXOVMS~~SPUV
UXTY~~LPNVNXLYJXIWGT~~KPMVMXLY!!TPURVUVXTYSYPXNVMSMP~~SPTRUUUXTY~~MPLTKU
JUHT!!UPXRYUYXWYVYTXRV~~UPWRXUXXWY~~PPRVRXPYOYMXKVJSJP~~OPQVQXPY~~JPITHU
```

TITLE

VARIABLE FONT PRESENTATION AID

ABSTRACT NO:

51/00-8040/0

```
GUET!!XTVQTPRPQQQSRVSXUYUYWXWWUWWX~~KQLRKRKQLPMPOQPSQVQXPYMYJWHT~~QQPS~~
OQQS~~RVQX~~QVSX!!TPVVUXTYSYQXOVMS~~SPUVUUXTY~~LPRb~~KPHYP↑RbSdUeVdVbU_S]
PELYIWGT!!UTTQRPPPNQNTOVRXTY~~PPOQOTPVRX~~TYRZQ\Q_RbTdUeWdWbV_S\PZLWIT~~
TYSZR\R_SbTd!!
```

TEKTRONIX 4051 PROGRAM

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
BINARY DATA FILE DUPLICATOR	51/00-8041/0
ORIGINAL DATE: February 1980 / REVISION DATE:	MEMORY REQUIREMENT: 6K + Data Storage
AUTHOR: R. Molins, Y. Juguet — National Polytechnic Institute, Grenoble, France	PERIPHERALS: None

ABSTRACT

Statements: 231

Files: 1 ASCII Program

The program copies the contents of a binary data file into another file, on the same tape or a different one, without using a peripheral device. The program reads the binary data from the file into memory, until the end of the file is reached or memory space is filled.

ASCII characters "/" and "#" are used as logical record separators. The program allows the user to change the separator character if it is already used in the data. The program then reads the binary data from the file into memory. Each logical record is processed one at a time by converting it to an ASCII character string. Each character, if not a separator, is linked to the previous one and so on until a separator is encountered. The separator is deleted after recognition of type of data involved and the resultant string is converted either into a string or numeric data. The string is then stored on the tape. Then it searches the target file, and writes each logical record after it recognizes and converts it into a string or a numeric variable. If necessary, this process is repeated until the complete file is duplicated.

The program also allows the user to modify the contents of the file during the process of duplication. The modification can occur in one of two ways:

1. Delete or change a logical record by addressing its assigned number, or
2. Systematically changing or deleting a specified logical record each time it is encountered in the data file.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

The program is designed to duplicate binary data files only; it will not process binary program files, or any file type other than binary data.

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATION	51/00-8041/0

METHOD

Reading Routine

Computation of available memory

Initialization of Z$ = " "

Recognition of the next logical record by TYP(0)

Reading of the record

Conversion to an ASCII string

Concatenation of a distinctive separator

Concatenation into Z$

Within available memory return to the next logical record

Filling memory without EOF occuring, the last record is located and control is passed back to the writing routine

On EOF control is passed back to the writing routine

Writing Routine

Search of target file

Determination of structure of the target file

Position of the tape either at the beginning of the file or the end

Analysis of the string Z$ character by character

each character, if not a separator, is linked to the previous and so on until a separator is encountered. The separator is deleted after recognition of type of data involved and the resultant string is converted either into a string or numeric data.

the string or data is stored on the tape.

Duplication ends when the whole string Z$ has been analysed and transfered onto the tape.

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

PROGRAM LOADING

Press AUTOLOAD if the program is located on the first file on the tape, otherwise:

FIND # (file number)

OLD

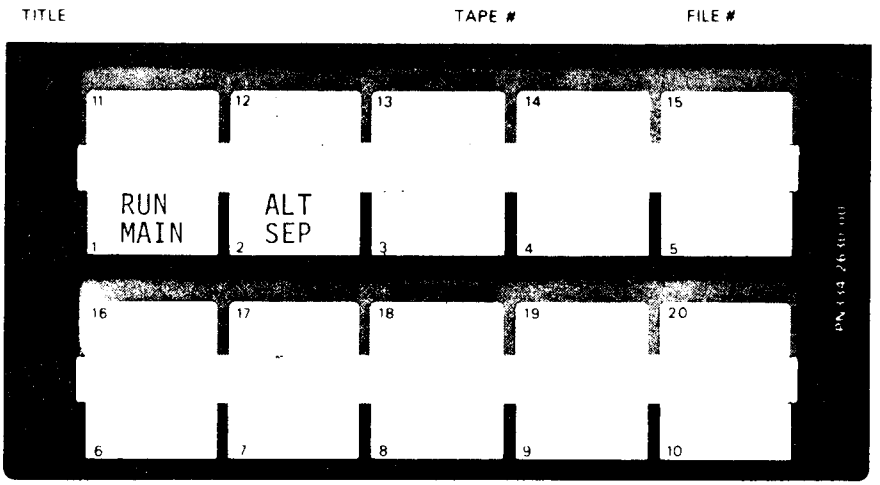
RUN

Respond to the questions as they are asked.

NOTE: The program will transfer only binary data files. If an ASCII file is attempted the program will abort and an error message will be displayed.

The target file must be pre-MARked or emptied before beginning the writing phase of the program.

TITLE	ABSTRACT NUMBER
BINARY DATA FILE DUPLICATOR	51/00-8041/0



UDK #1	FUNCTION
1	Runs the main program section - transfer of the data from one file to another as it is.
2	Alternate separator search routine - allows an alternate record separator to be specified

BINARY FILE DUPLICATION

```
------------------
| LOAD SOURCE TAPE |
------------------
```

THIS PROGRAM NEEDS LOGIC RECORD SEPARATORS. THEY ARE ASCII CHARACTERS "~" AND "#".

THEY MUST BE ABSENT FROM SOURCE FILE.
IF NOT, YOU MUST CHOOSE ALTERNATE SEPARATORS.
I MAY HELP YOU,.....PRESS UDK No 2

IF THEY ARE ABSENT FROM SOURCE FILE, PRESS UDK No 1

INPUT SOURCE FILE NUMBER :

THIS SEARCH MAY BE TEDIOUS AND TIME CONSUMING.
BE PATIENT, I'LL LET YOU KNOW WHEN I'VE COMPLETED MY JOB.

IS THE TARGET FILE ON THE SAME TAPE ?

TAKE CARE ! THE TARGET FILE MUST BE EMPTY, OTHERWISE THE CONTENT OF SOURCE FILE WILL BE WRITTEN AFTER ITS OWN CONTENT.

INPUT TARGET FILE NUMBER :

```
                    WILL YOU CHANGE SOME LOGIC RECORDS ?Y

WILL YOU REPLACE SYSTEMATICALLY ONE LOGIC RECORD BY ANOTHER ONE ? N


THEREFORE YOU WILL REPLACE SOME LOGIC RECORDS GIVING ME THEIR SERIAL
NUMBER ON THE TAPE !

I AM BEGINNING TO READ THE TAPE FILE AND I'LL GIVE YOU EACH LOGIC RECORD
WITH ITS NUMBER FILE;

KEEP IT IN MIND,I'LL ASK YOU LATER FOR ITS REPLACEMENT.


  1     39.912#
  2     44.9653333333#
  3     2#
  4     0.0239999999994#
  5     9.984#
  6     1#
  7     9.984#
  8     0#
  9     1#
  10     0#
  11     -9.984#
  12     1#
   .
   .
   .
   .
```

```
I JUST COMPLETED RECORDING SOURCE FILE
AND I'M STARTING ITS TRANSFER ONTO THE TARGET FILE.

HOW MANY LOGIC RECORDS ARE TO BE CHANGED ?



 ENTER THE NUMBER OF THE RECORD TO BE CHANGED :



 LOGIC RECORD No 1 IS :  39.912

 ENTER YOUR CORRECTION !

 IF YOU ENTER "~",THE FILE WILL BE CLOSED
                  AND FOLLOWING RECORDS WILL BE LOST.

 TAKE CARE, CORRECTION MUST BE A NUMERIC VALUE
```

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

DATA STORAGE

Variable	Used to Store	Type
A	Numeric data	Simple
A$	String for reading or writing operations	String
B	Number of logical records to be changed	Simple
B$	String of used separators	String
C	Number file of logical records to be changed	Array (B)
C$	Character analysed from Z$	String
D	Result from a test	Array (13)
D1	Result from a test	Simple
D$	String of available separators	String
E	Defines replacing mode	Simple
F	Source file or target file	Simple
I	Loop variable	Simple
I1	Loop variable	Simple
J	Source file number	Simple
K	Target file number	Simple
L	Number file of present logic record	Simple
M	Available memory	Simple
T1	TYP(0) from source file	Simple
T2	TYP(0) from target file	Simple
T$	Answer to "Same tape?"	String
U	Position of C$ in U$	Simple
U$	String to be changed systematically	String
V$	Its substitute	String
W	Position of W$ in "YN"	Simple
W$	Answer Y or N	String
X$	String separator character	String
Y$	Numeric variable separator character	String
Z	Length of Z$	Simple
Z$	Working string	String (M)

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

```
1 GO TO 100
4 GO TO F OF 1150,920,270
8 GOSUB 1760
9 GO TO 310

19 REM *** 51/00-8041/0 *** COPY OF BINARY DATA FILES ***
20 REM ****** WITH OPTION FOR CHANGE SOME LOGIC RECORDS ******

50 INPUT W$
51 W$=SEG(W$,1,1)
52 W=POS("YN",W$,1)
53 IF W=0 THEN 55
54 RETURN
55 PRINT """"GGVALID ANSWERS ARE ""YES"" OR ""NO""""""
56 GO TO 50
60 FOR I=1 TO 10
65     PRINT "GG";
70     FOR I1=1 TO 10
75     NEXT I1
80 NEXT I
85 RETURN
100 PAGE
110 PRINT " GGGG                    BINARY FILE DUPLICATION J"
120 INIT
130 PRINT "I ------------------"
140 PRINT "I| LOAD SOURCE TAPE |"
150 PRINT "I ____________________JJ"
160 PRINT "THIS PROGRAM NEEDS LOGIC RECORD SEPARATORS. THEY ARE ASCII";
170 PRI " CHARACTERSJ I""~"" AND ""#"". JJJJTHEY MUST BE ABSENT FROM ";
180 PRINT "SOURCE FILE._IF NOT, YOU MUST CHOOSE ALTERNATE SEPARATORS."
190 PRINT "I MAY HELP YOU,.....PRESS UDK No 2"
200 PRINT "JJIF THEY ARE ABSENT FROM SOURCE FILE, PRESS UDK No 1"
210 F=3
220 X$="~"
230 Y$="#"
240 J=0
```

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

```
250 E=0
260 END
270 PAGE

280 REM            ******  INITALIZE PARAMETERS  ******

290 PRINT "JJJGINPUT SOURCE FILE NUMBER : ";
300 INPUT J
310 PRINT "JJGIS THE TARGET FILE ON THE SAME TAPE ?";
320 GOSUB 50
330 T$=W$
340 PRINT "JJTAKE CARE ! THE TARGET FILE MUST BE EMPTY, OTHERWISE ";
350 PRINT "THE CONTENT OF SOURCE FILE WILL BE WRITTEN AFTER ITS OWN ";
360 PRINT "CONTENT."
370 PRINT "JJGINPUT TARGET FILE NUMBER : ";
380 INPUT K
390 PRINT "LJJJJIWILL YOU CHANGE SOME LOGIC RECORDS ?";
400 GOSUB 50
410 IF W=2 THEN 590
420 PRINT "JWILL YOU REPLACE SYSTEMATICALLY ONE LOGIC RECORD ";
430 PRINT "BY ANOTHER ONE ? ";
440 GOSUB 50
450 E=W
460 IF E=2 THEN 520
470 PRINT "JJWHAT IS THE LOGIC RECORD TO BE REPLACED ?";
480 INPUT U$
490 PRINT "J        .........................BY  ?";
500 INPUT V$
510 GO TO 590
520 PRI "JJTHEREFORE YOU WILL REPLACE SOME LOGIC RECORDS GIVING ME THE";
530 PRINT "IR SERIAL NUMBER ON THE TAPE !J"
540 PRI "I AM BEGINNING TO READ THE TAPE FILE AND I'LL GIVE YOU EACH ";
550 PRINT "LOGIC RECORDWITH ITS NUMBER FILE; JKEEP IT IN MIND,I'LL ";
560 PRINT "ASK YOU LATER FOR ITS REPLACEMENT."
570 FOR I=1 TO 2500
580 NEXT I
```

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

```
590 M=MEMORY-500
600 DIM Z$(M)
610 Z$=""
620 L=1
630 F=1
640 PAGE

650 REM      ****** SOURCE FILE RECORDING *****

660 FIND J
670 T1=TYP(0)
680 GOSUB T1 OF 750,1480,1240,1370
690 IF E<▮2 THEN 710
700 PRINT L;"   ";A$
710 L=L+1
720 Z$=REP(A$,1+LEN(Z$),0)
730 IF LEN(Z$)▮M-75 THEN 780
740 GO TO 670
750 PRINT "I JUST COMPLETED RECORDING SOURCE FILE   GGGGG"
760 PRINT "AND I'M STARTING ITS TRANSFER ONTO THE TARGET FILE."
770 GO TO 800
780 PRINT "I'VE READ A PART OF SOURCE FILE, BUT MY MEMORY IS FULL";
790 PRINT " JJI MUST COPY THIS PART BEFORE GOING AHEAD !"
800 IF E<▮2 THEN 890
810 PRINT "JJJHOW MANY LOGIC RECORDS ARE TO BE CHANGED ? ";
820 GOSUB 60
830 INPUT B
840 DIM C(B)
850 FOR I1=1 TO B
860     PRINT "LJJJJJ  ENTER THE FILE NUMBER OF THE RECORD TO BE CHANGED :";
870     INPUT C(I1)
880 NEXT I1

890 REM  ******   RECORDING ONTO THE TARGET FILE  ******

900 F=2
```

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

```
>10 IF T$="N" THEN 1520
920 FIND K
930 T2=TYP(0)
940 IF T2<>0 THEN 1160
950 FIND K
960 A$=""
970 L=0
980 B$=X$&Y$
990 Z=LEN(Z$)
1000 FOR I=1 TO Z
1010     C$=SEG(Z$,I,1)
1020     U=POS(B$,C$,0)
1030     GOSUB U OF 1400,1280
1040     A$=A$&C$
1050 NEXT I
1060 CLOSE
1070 IF T1<>1 THEN 1110
1080 PRINT "LJJJJJJIMY JOB IS OVERJJJJJI GOOD BYE !"
1090 GOSUB 60
1100 END
1110 Z$=""

1120 REM ****** POSITION OF R/W HEAD ON EOF ******

1130 F=1
1140 IF T$="N" THEN 1540
1150 FIND J
1160 FOR I=1 TO L-1
1170     IF F=2 AND I=1 THEN 1190
1180     T2=TYP(0)
1190     GOSUB T2 OF 960,1480,1240,1370
1200     DELETE A$
1210 NEXT I
1220 GO TO F OF 670,960

1230 REM *****  SUBROUTINE READ OR WRITE DATA  ******
```

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

```
1240 READ @33:A
1250 A$=STR(A)
1260 A$=A$&Y$
1270 RETURN
1280 L=L+1
1290 GOSUB E OF 1580,1610
1300 IF A$<▮"" THEN 1320
1310 GO TO 1360
1320 A=VAL(A$)
1330 A$=""
1340 C$=""
1350 WRITE A
1360 RETURN
1370 READ @33:A$
1380 A$=A$&X$
1390 RETURN
1400 L=L+1
1410 GOSUB E OF 1580,1610
1420 IF A$<▮"" THEN 1440
1430 GO TO 1470
1440 WRITE A$
1450 A$=""
1460 C$=""
1470 RETURN
1480 PRINT "JJTHIS FILE IS NOT A BINARY DATA FILE. JI'VE NOT BEEN PRO";
1490 PRINT "GRAMMED FOR ITS TRANSFER !"
1500 GOSUB 60
1510 END
1520 FOR I=1 TO 200
1530 NEXT I
1540 PRINT "LJJJIPLEASE CHANGE YOUR TAPE !"
1550 PRINT "JJJ          AND GIVE ME CONTROL BY PRESSING UDK No 1"
1560 GOSUB 60
1570 END
1580 IF A$<▮U$ THEN 1600
```

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

```
1590 A$=V$
1600 RETURN
1610 FOR I1=1 TO B
1620     IF L<▮C(I1) THEN 1730
1630     PAGE
1640     PRINT "JJJJLOGIC RECORD No ";L;" IS : ";A$
1650     PRINT "JJENTER YOUR CORRECTION !"
1660     PRINT "JJIF YOU ENTER """;X$;""",THE FILE WILL BE CLOSED JIAND ";
1670     PRINT "FOLLOWING RECORDS WILL BE LOST."
1680     IF U=1 THEN 1710
1690     PRINT "JJTAKE CARE, CORRECTION MUST BE A NUMERIC VALUE ";
1700     MOVE 10,40
1710     INPUT A$
1720     IF A$=X$ THEN 1060
1730 NEXT I1
1740 RETURN

1750 REM ******* SEARCH OF ALTERNATE SEPARATORS ******

1760 PAGE
1770 MOVE 20,50
1780 PRINT "JJJGINPUT SOURCE FILE NUMBER : ";
1790 INPUT J
1800 PAGE
1810 MOVE 0,50
1820 PRINT "THIS SEARCH MAY BE TEDIOUS AND TIME CONSUMING."
1830 PRINT "BE PATIENT, I'LL LET YOU KNOW WHEN I'VE COMPLETED MY JOB."
1840 D$="~#$%&':*@<▮!"
1850 DIM D(12)
1860 D=1
1870 FIND J
1880 T1=TYP(0)
1890 GOSUB T1 OF 2020,1480,1980,2000
1900 IF T1=3 THEN 1880
1910 FOR I=1 TO LEN(A$)
1920     C$=SEG(A$,I,1)
```

TITLE	ABSTRACT NO:
BINARY DATA FILE DUPLICATOR	51/00-8041/0

```
1930      D1=POS(D$,C$,1)
1940      IF D1=0 THEN 1960
1950      D(D1)=0
1960 NEXT I
1970 GO TO 1880
1980 READ @33:A
1990 RETURN
2000 READ @33:A$
2010 RETURN
2020 IF SUM(D)<>1 THEN 2060
2030 PRINT "I'VE NOT FOUND VALID SEPARATORS. YOU MUST RECALL LINE 1840";
2040 PRINT " AND REPLACE SOME OF THE POSSIBLE SEPARATORS BY OTHER ONES"
2050 END
2060 I=1
2070 IF D(I)=0 THEN 2110
2080 X$=SEG(D$,I,1)
2090 D(I)=0
2100 GO TO 2130
2110 I=I+1
2120 GO TO 2070
2130 I=1
2140 IF D(I)=0 THEN 2180
2150 Y$=SEG(D$,I,1)
2160 D(I)=0
2170 GO TO 2200
2180 I=I+1
2190 GO TO 2140
2200 GOSUB 60
2210 GO TO 310
```

TEKTRONIX®

TAPE 19 F 43-45

4051

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
Overlay Drawing Program	51/00-9537/0
ORIGINAL DATE: February 1980 / REVISION DATE:	MEMORY REQUIREMENT: 16K
AUTHOR: LeRoy Nollette, Tektronix, Inc., Wilsonville, OR	PERIPHERALS: 4662 Plotter

ABSTRACT

Statements: 250

Files: 1 ASCII Program
2 ASCII Data (Sample Overlays)
Optional - Pre-MARked data files

The program draws an overlay on the 4662 Plotter that can be cut out and placed over the User-Definable Keys. Key descriptions may be entered from the keyboard. Data may be saved on a pre-MARked data file and redrawn at a later date.

The program may be modified (one line of code) to draw a large copy of the overlay and then reduce it on a copy machine having reduction capabilities.

By changing one line of code the user may preview the overlay on the screen.

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

DESCRIPTION

The program allows the user to create a very nice, printed overlay using the 4050 Series Desktop Computer and the 4662 Plotter. The printed overlay may be cut out and glued to an overlay.

The titles and key descriptions may be saved to a pre-MARked file, allowing the user to print the overlay again at a later time.

OPERATING INSTRUCTIONS

The 4662 Plotter should be on and set to the default page size (same as when turned on). The device address should be set at 1 (factory setting). If the device address is not 1 change the value of T in line 120 to the correct address.

Load a piece of paper and remove the pen cap.

Press AUTOLOAD (if the program is located on the first file) or:

```
FIND #  (file number)
OLD
RUN
```

Respond to the questions as they are asked.

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

Program Execution

```
OVERLAY DRAWING PROGRAM

(1)TAPE
(2)KEYBOARD
(3)BLANK

 SELECT SOURCE OF KEY LABELS
```

(1) TAPE

Allows the user to draw an overlay that has been created and saved on a pre-MARked tape file.

(2) KEYBOARD

Allows the user to enter from the keyboard the titles and key labels.

(3) BLANK

Allows the user to draw a blank overlay.

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

(1) TAPE

```
OVERLAY DRAWING PROGRAM

(1)TAPE
(2)KEYBOARD
(3)BLANK

 SELECT SOURCE OF KEY LABELS 1
INPUT FILE NUMBER
```

Input the file number of the overlay to be drawn.

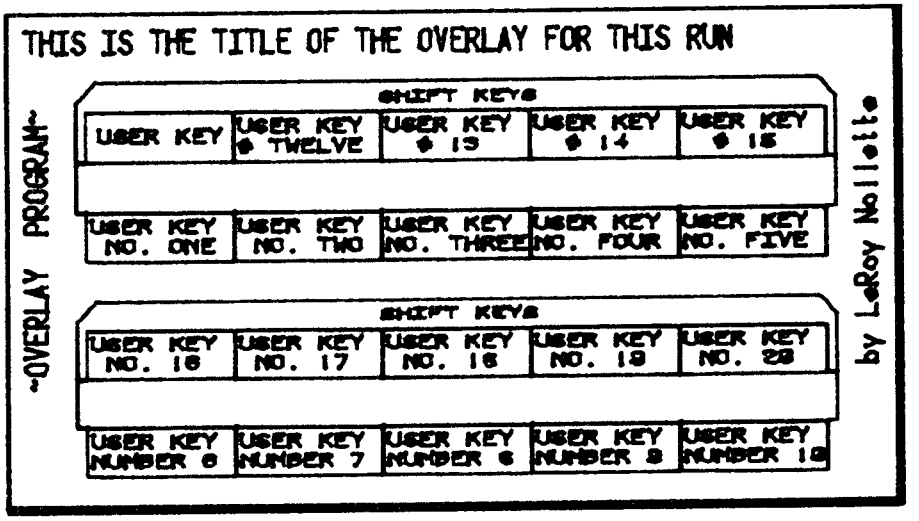
The plotter pen will move to the lower left corner.

The pen may be positioned to any location. Be sure to leave sufficient room to the right and above to draw the overlay. The pen position will be the lower left corner of the overlay.

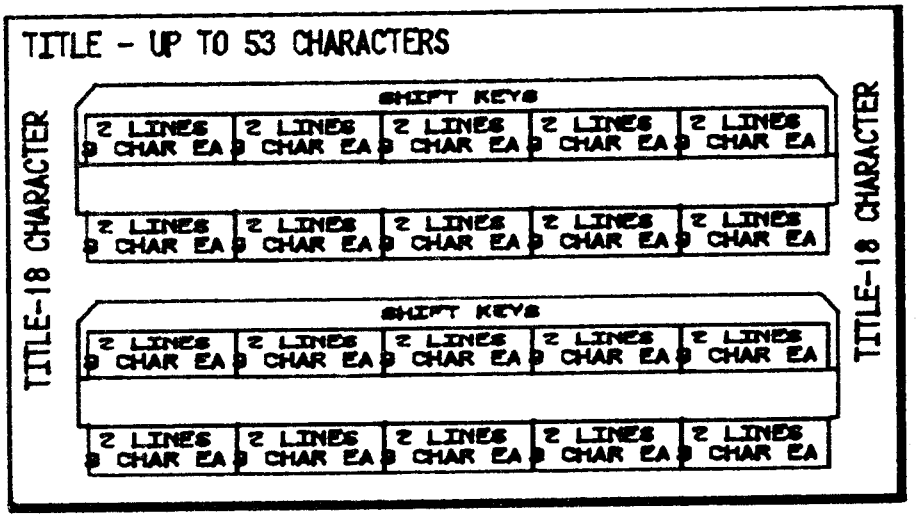
When the pen is in position press the CALL button on the plotter. The overlay will then be drawn.

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

Example of file 2



Example of file 3



TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

(2) KEYBOARD

```
OVERLAY DRAWING PROGRAM

(1)TAPE
(2)KEYBOARD
(3)BLANK

 SELECT SOURCE OF KEY LABELS 2
PLEASE ENTER THE TITLE OF THE OVERLAY
_____________________________________________
ENTER THE TITLE ON LEFT EDGE OF OVERLAY __________________
ENTER THE TITLE ON RIGHT EDGE OF OVERLAY __________________

HOW MANY LINES IN DESCRIPTION OF KEY 11 2
ENTER DESCRIPTION OF LINE 1 FOR KEY 11 _________
ENTER DESCRIPTION OF LINE 2 FOR KEY 11 _________

HOW MANY LINES IN DESCRIPTION OF KEY 12 1
ENTER DESCRIPTION OF LINE 1 FOR KEY 12 _________

HOW MANY LINES IN DESCRIPTION OF KEY 13 0
```

The title of the overlay may contain up to 53 characters.

The title on the left edge may contain up to 18 characters.

The title on the right edge may contain up to 18 characters.

The key labels may contain up to 2 lines of 9 characters each.

The program asks for the number of lines of data for each key; if no label is desired enter Ø.

```
DO YOU WANT THE DATA STORED TO TAPE NO
```

The plotter pen will move to the lower left corner.

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

The pen may be positioned to any location. Be sure to leave sufficient room to the right and above to draw the overlay. The pen position will be the lower left corner of the overlay.

When the pen is in position, press the CALL button on the plotter. The overlay will then be drawn.

DO YOU WANT THE DATA STORED TO TAPE YES
INPUT FILE NUMBER

Input the file number of a pre-MARked file for the data to be saved.

The plotter pen will move to the lower left corner.

The pen may be positioned to any location. Be sure to leave sufficient room to the right and above to draw the overlay. The pen position will be the lower left corner of the overlay.

When the pen is in position press the CALL button on the plotter. The overlay will then be drawn.

NOTE: Files should be marked for 1000 bytes.

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

(3) BLANK

OVERLAY DRAWING PROGRAM

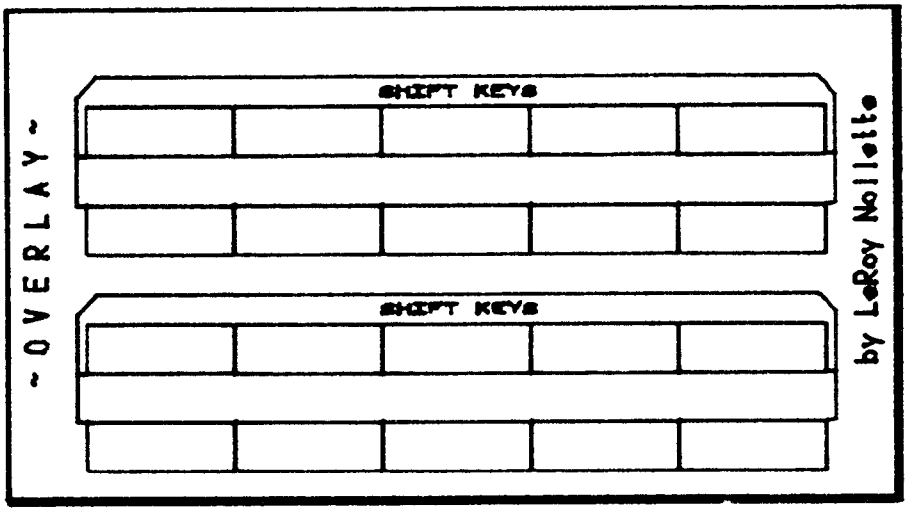
(1)TAPE
(2)KEYBOARD
(3)BLANK

SELECT SOURCE OF KEY LABELS 3

The plotter pen will move to the lower left corner.

The pen may be positioned to any location. Be sure to leave sufficient room to the right and above to draw the overlay. The pen position will be the lower left corner of the overlay.

When the pen is in position press the CALL button on the plotter. The overlay will then be drawn.



TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

SPECIAL FEATURES

Screen Preview

The overlay may be previewed on the screen. The character rotation and the variable character sizes are not available on the screen. A full screen version will be displayed.

Change:

```
120 T=1  to  120 T=32
```

Enlarge Overlay

The overlay may be enlarged and then reduced on a copy machine with reduction capabilities.

Line 700 is currently:

```
700 S1=1
```

If a copy machine reduces 64%, change line 700 to:

```
700 S1=.64
```

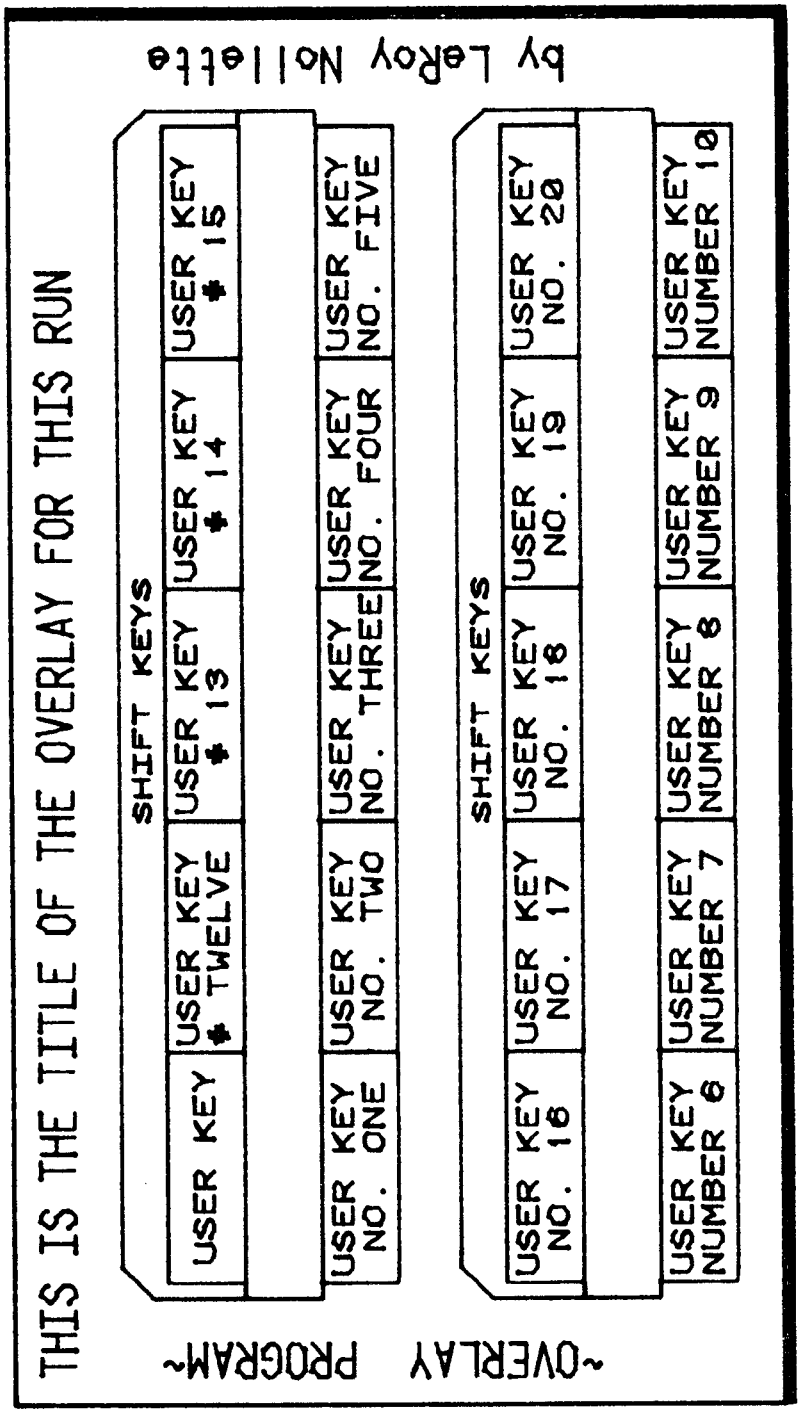
the resulting plot would be 64% larger and after being reduced would give a clear picture the proper size for an overlay.

TITLE

Overlay Drawing Program

ABSTRACT NO:

51/00-9537/0

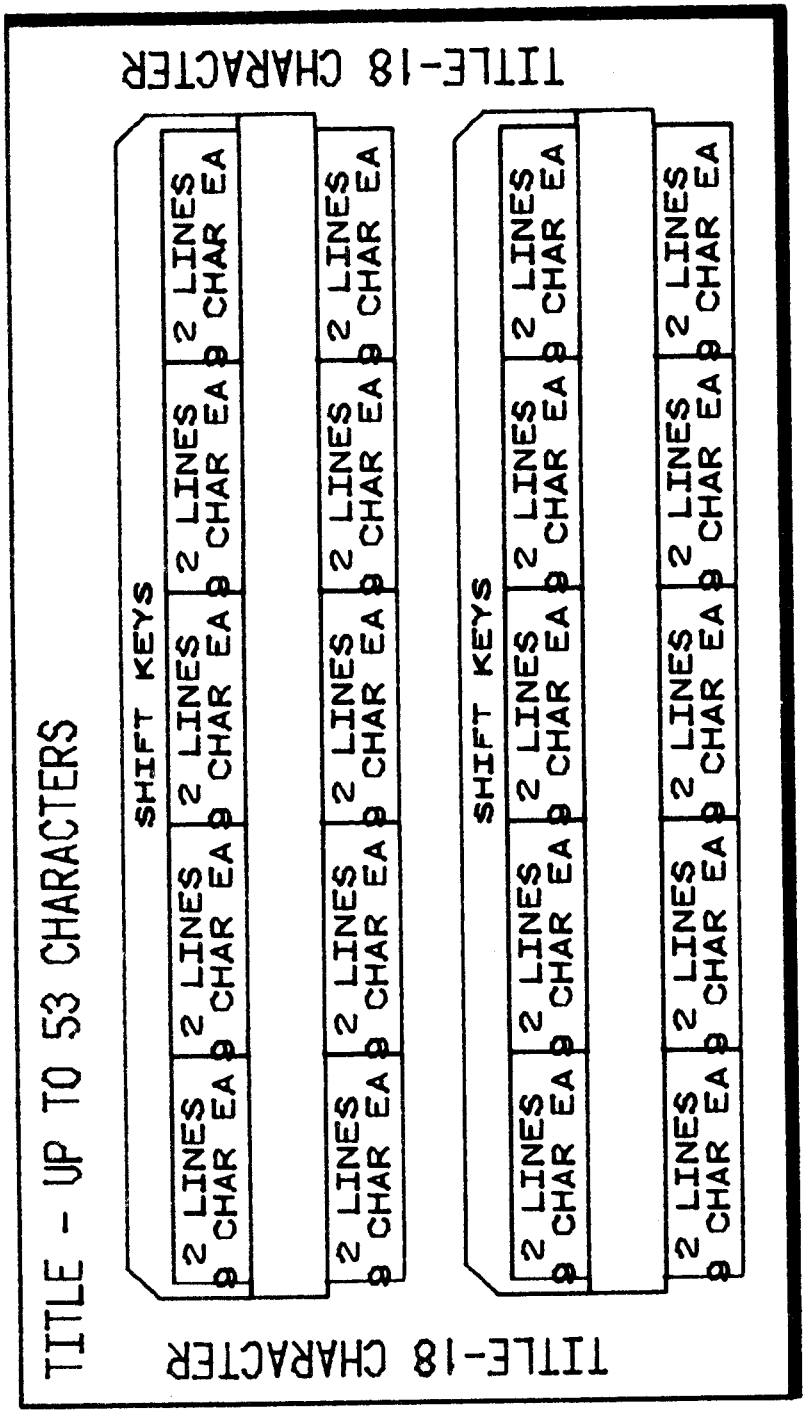


TITLE

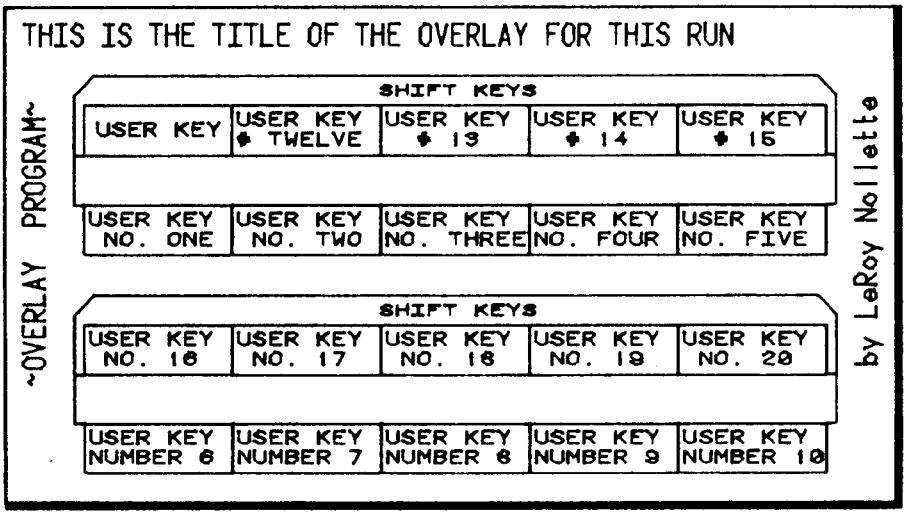
Overlay Drawing Program

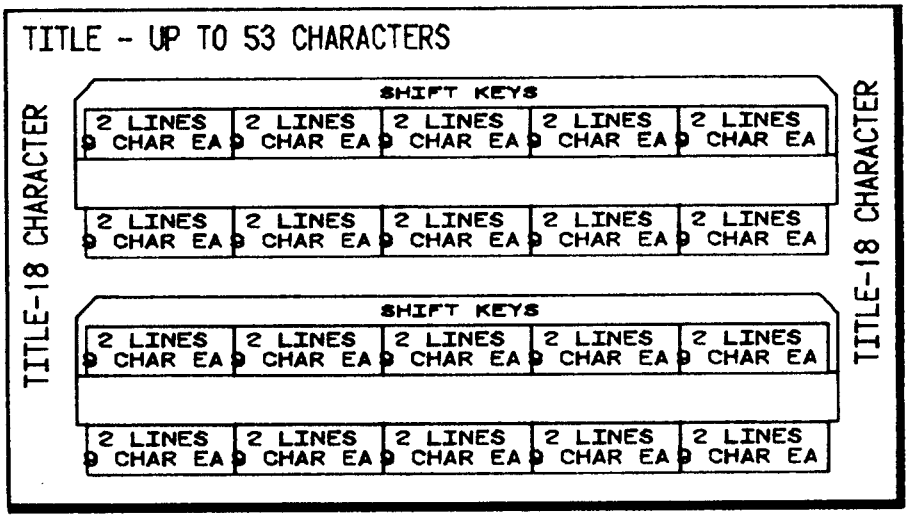
ABSTRACT NO:

51/00-9537/0



TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0





TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

```
1 REM ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2 REM $   OVERLAY DRAWING PROGRAM                                          $
3 REM $                              by LeRoy Nollette                    $
4 REM $                                  Tektronix, Inc.                  $
5 REM $                                   Beaverton, OR 97077             $
6 REM ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

100 INIT
110 SET DEGREES
120 T=1
130 PAGE
140 K=130+20*(T=1)
150 VIEWPORT 0,K,0,100
160 WINDOW 0,K,0,100
170 DIM A$(800),B$(200)

180 REM TITLE

190 A$="<"

200 REM LEFT SIDE VERTICAL

210 A$=A$&"~ O V E R L A Y ~<"

220 REM RIGHT SIDE VERTICAL

230 A$=A$&"by LeRoy Nollette<"

240 REM KEY 11

250 A$=A$&"<"

260 REM KEY 12

270 A$=A$&"<"
```

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

```
280 REM KEY 13
290 A$=A$&"<"
300 REM KEY 14
310 A$=A$&"<"
320 REM KEY 15
330 A$=A$&"<"
340 REM KEY 1
350 A$=A$&"<"
360 REM KEY 2
370 A$=A$&"<"
380 REM KEY 3
390 A$=A$&"<"
400 REM KEY 4
410 A$=A$&"<"
420 REM KEY 5
430 A$=A$&"<"
440 REM KEY 16
450 A$=A$&"<"
```

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

```
460 REM KEY 17
470 A$=A$&"<"
480 REM KEY 18
490 A$=A$&"<"
500 REM KEY 19
510 A$=A$&"<"
520 REM KEY 20
530 A$=A$&"<"
540 REM KEY 6
550 A$=A$&"<"
560 REM KEY 7
570 A$=A$&"<"
580 REM KEY 8
590 A$=A$&"<"
600 REM KEY 9
610 A$=A$&"<"
620 REM KEY 10
630 A$=A$&"<"
```

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

```
640 GOSUB 1920

650 REM ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
660 REM S1 is the scaling factor (usually 1)
670 REM if you have a coping machine that reduces 70%
680 REM you can enter S1=.70 to produce large copy to be reduced.
690 REM ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

700 S1=1
710 IF T=1 THEN 780
720 X=0
730 Y=5
740 S1=0.35
750 MOVE @T:X,Y
760 SCALE S1,S1
770 GO TO 810
780 PRINT @T:"G";
790 PRINT @T,21:5,5
800 INPUT @T,27:X,Y
810 MOVE @T:X,Y
820 SCALE S1,S1
830 X=0
840 Y=0
850 FOR I=0 TO 0.3 STEP 0.1
860 RDRAW @T:0,25
870 RDRAW @T:45+I,0
880 RDRAW @T:0,-(25+I)
890 RDRAW @T:-(45+I),0
900 RDRAW @T:0,I
910 NEXT I
920 RMOVE @T:1,23
930 S=1/S1
940 PRINT @T,17:0.8*S,1.8*S
950 PRINT @32,17:4
960 P=0
970 GOSUB 1870
```

TITLE

Overlay Drawing Program

ABSTRACT NO:

51/00/9537/0

```
980 PRINT @T:B$;
990 MOVE @T:X+2,Y+6
1000 GOSUB 1870
1010 IF T=1 THEN 1120
1020 INPUT @T,24:X5,Y5
1030 PRINT @T,21:X5,Y5+LEN(B$)*3-9
1040 INPUT @T,24:X5,Y5
1050 FOR J=1 TO LEN(B$)
1060 PRINT @T,21:X5,Y5
1070 C$=SEG(B$,J,1)
1080 PRINT @T:C$;
1090 Y5=Y5-3
1100 NEXT J
1110 GO TO 1140
1120 PRINT @T,25:90
1130 PRINT @T:B$;
1140 GOSUB 1870
1150 MOVE @T:X+44,Y+7
1160 IF T=1 THEN 1270
1170 INPUT @T,24:X5,Y5
1180 PRINT @T,21:X5,Y5+LEN(B$)*3-9
1190 INPUT @T,24:X5,Y5
1200 FOR J=1 TO LEN(B$)
1210 PRINT @T,21:X5,Y5
1220 C$=SEG(B$,J,1)
1230 PRINT @T:C$;
1240 Y5=Y5-3
1250 NEXT J
1260 GO TO 1290
1270 PRINT @T:B$;
1280 PRINT @T,17:0.8*S,0.8*S
1290 PRINT @T,25:0
1300 PRINT @32,17:2
1310 X=X+4
1320 FOR Y1=Y+4 TO Y+15 STEP 11
1330 FOR X1=X TO X+30 STEP 7.5
```

TITLE

Overlay Drawing Program

ABSTRACT NO:

51/00-9537/0

```
1340 MOVE @T:X1,Y1+2.5
1350 DRAW @T:X1+7.5,Y1+2.5
1360 DRAW @T:X1+7.5,Y1+2.5+2.5
1370 DRAW @T:X1,Y1+2.5+2.5
1380 DRAW @T:X1,Y1+2.5
1390 MOVE @T:X1,Y1
1400 DRAW @T:X1,Y1-2.5
1410 DRAW @T:X1+7.5,Y1-2.5
1420 DRAW @T:X1+7.5,Y1
1430 DRAW @T:X1,Y1
1440 NEXT X1
1450 NEXT Y1
1460 FOR Y1=Y+4 TO Y+15 STEP 11
1470 MOVE @T:X,Y1
1480 RDRAW @T:-0.5,0
1490 RDRAW @T:0,2.5
1500 RDRAW @T:0.5,0
1510 RDRAW @T:-0.5,0
1520 RDRAW @T:0,3
1530 RDRAW @T:1,1
1540 RDRAW @T:36.5,0
1550 RDRAW @T:1,-1
1560 RDRAW @T:0,-3
1570 RDRAW @T:-0.5,0
1580 RDRAW @T:0.5,0
1590 RDRAW @T:0,-2.5
1600 RDRAW @T:-0.5,0
1610 MOVE @T:X+15,Y1+5.5
1620 PRINT @T:"SHIFT KEYS";
1630 NEXT Y1
1640 PRINT @T,17:0.8*S,1*S
1650 DATA 19,14,8,3
1660 RESTORE 1650
1670 FOR J=1 TO 4
1680 READ Y1
1690 Y1=Y+Y1
```

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

```
1700 FOR X1=X TO X+30 STEP 7.5
1710 MOVE @T:X1+3.9,Y1
1720 GOSUB 1870
1730 L1=LEN(B$)
1740 P1=POS(B$,">",1)
1750 IF P1=0 THEN 1790
1760 C$=SEG("JHHHHHHHHHHHHHHHH",1,INT(P1/2+(L1-P1)/2+0.5))
1770 B$=REP(C$,P1,1)
1780 GO TO 1810
1790 RMOVE @T:0,-0.5
1800 P1=L1
1810 RMOVE @T:-(P1/2*0.8),0
1820 PRINT @T:B$;
1830 NEXT X1
1840 NEXT J
1850 X=X-4
1860 END

1870 REM SELECT B$ ROUTINE

1880 Q=P+1
1890 P=POS(A$,"<",Q)
1900 B$=SEG(A$,Q,P-Q)
1910 RETURN

1920 REM CREATE THE OVERLAY TEXT

1930 PRINT "OVERLAY DRAWING PROGRAMJ_(1)TAPE_(2)KEYBOARD "
1940 PRINT "(3)BLANK _J SELECT SOURCE OF KEY LABELS ";
1950 INPUT A1
1960 GO TO A1 OF 1980,2030,2020
1970 GO TO 1930
1980 PRINT "INPUT FILE NUMBER ";
1990 INPUT F
2000 FIND F
2010 INPUT @33:A$
```

TITLE	ABSTRACT NO:
Overlay Drawing Program	51/00-9537/0

```
2020 GO TO 2500
2030 PRINT "PLEASE ENTER THE TITLE OF THE OVERLAY "
2040 PRINT USING "53""_"",53""H""S":
2050 INPUT A$
2060 A$=A$&"<"
2070 PRINT "ENTER THE TITLE ON LEFT EDGE OF OVERLAY ";
2080 PRINT USING "18""_"",18""H""S":
2090 INPUT B$
2100 B$=B$&"<"
2110 A$=A$&B$
2120 PRINT "ENTER THE TITLE ON RIGHT EDGE OF OVERLAY ";
2130 PRINT USING "18""_"",18""H""S":
2140 INPUT B$
2150 B$=B$&"<"
2160 A$=A$&B$
2170 DATA 11,12,13,14,15,1,2,3,4,5,16,17,18,19,20,6,7,8,9,10
2180 RESTORE 2170
2190 FOR I=1 TO 20
2200 READ A1
2210 PRINT "JHOW MANY LINES IN DESCRIPTION OF KEY ";A1;" ";
2220 INPUT J1
2230 IF J1=0 THEN 2520
2240 IF J1>2 THEN 2210
2250 B$=""
2260 DIM J2(2)
2270 C$=""
2280 FOR J=1 TO J1
2290 PRINT "ENTER DESCRIPTION OF LINE ";J;" FOR KEY ";A1;" ";
2300 PRINT USING "9""_"",9""H""S":
2310 INPUT C$
2320 J2(J)=LEN(C$)
2330 C$=C$&">"
2340 B$=B$&C$
2350 NEXT J
2360 B$=REP("<",LEN(B$),1)
2370 A$=A$&B$
```

TITLE

Overlay Drawing Program

ABSTRACT NO:

51/00-9537/0

```
2380 NEXT I
2390 PRINT "DO YOU WANT THE DATA STORED TO TAPE ";
2400 INPUT K$
2410 IF LEN(K$)=0 THEN 2500
2420 K$=SEG(K$,1,1)
2430 GO TO POS("YN",K$,1)+1 OF 2390,2450,2500
2440 GO TO 2390
2450 PRINT "INPUT FILE NUMBER ";
2460 INPUT F
2470 FIND F
2480 PRINT @33:A$
2490 CLOSE
2500 PRINT @T:"L";
2510 RETURN
2520 B$="<"
2530 GO TO 2370
```

TEKTRONIX

APPLICATIONS LIBRARY PROGRAM

TITLE	ABSTRACT NUMBER
ENHANCED PIE CHART	51/07-9539/0
ORIGINAL DATE / REVISION DATE	MEMORY REQUIREMENT: 32K
AUTHOR: Isaac C. Arthur, Jr. — U.S. Dept. of HUD, Washington, D.C.	PERIPHERALS: See Below

ABSTRACT

Peripherals: 4907 File Manager
4952 Joystick for screen display
or 4662 Plotter

Statements: 1249

Files: 7 Binary Program
Requires data files

The program draws a pie chart on either a 4050 Screen or 4662 Plotter. Data for the chart may be entered directly through the keyboard, read from tape or read from disk.

Maximum of 3 title lines; maximum of 64 characters per title line
Maximum of 10 segments
Maximum of 25 characters per segment label
Segment values may be those entered by the user or computed as percentages of the total
Segment separation options:
- All separated
- One separated
- None separated

User positions segment labels and values
Segment labels and values may be inside, outside or not displayed
Segments drawn on the Plotter may be multi-colored
10 User-Definable Keys manipulate the data:
- Save Plot (tape or disk)
- Display Data (titles, segment text and values)
- One Color Plot (4662)
- Multi-Color Plot (4662)
- Screen Plot
- Title Change
- Segment Text and Value Change

The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc., assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TITLE

ENHANCED PIE CHART

ABSTRACT NO

51/07-9539/0

Add Segments
Delete Segments
Chart Border (draws page borders)
End
Restart

A main program calls overlays from the 4907 File Manager.

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

METHOD

The program prompts the user for the source of chart data. Depending on the user response, the program reads the data from tape or disk or prompts the user for the required data to be entered in the following order:

1. Titles
2. Number of segments
3. Segment values and labels

Each data entered by the user is edited for reasonableness and compliance to program limitations. Changes to data in memory are effected through the User-Definable Keys.

The pie chart may be drawn on the 4050 Screen or 4662 Plotter; each output being the function of a subroutine.

Drawing on the 4050 Screen

The program prompts the user for the following:

1. Segment labels printing option
2. Symbol to be printed with segment values.
3. Values to be printed, raw user input or calculated percentage of segment to total pie
4. Percentage of window to be allocated to the circle
5. Segment separation option and percentage of separation if any
6. Positioning of titles if any

All the required instructions for the positioning of the cursor for segment labels and pointers where required are displayed on the screen and the user is given the option of making a hard copy of the instructions.

The pie chart is drawn on the screen segment by segment. The attention bell is rung after each segment has been drawn and the current cursor position is displayed to assist the user to position the cursor for segment label and pointer where required.

A hard copy of the complete pie chart is made for the user by the program before any further instructions are displayed on the screen.

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

Drawing on the 4662 Plotter

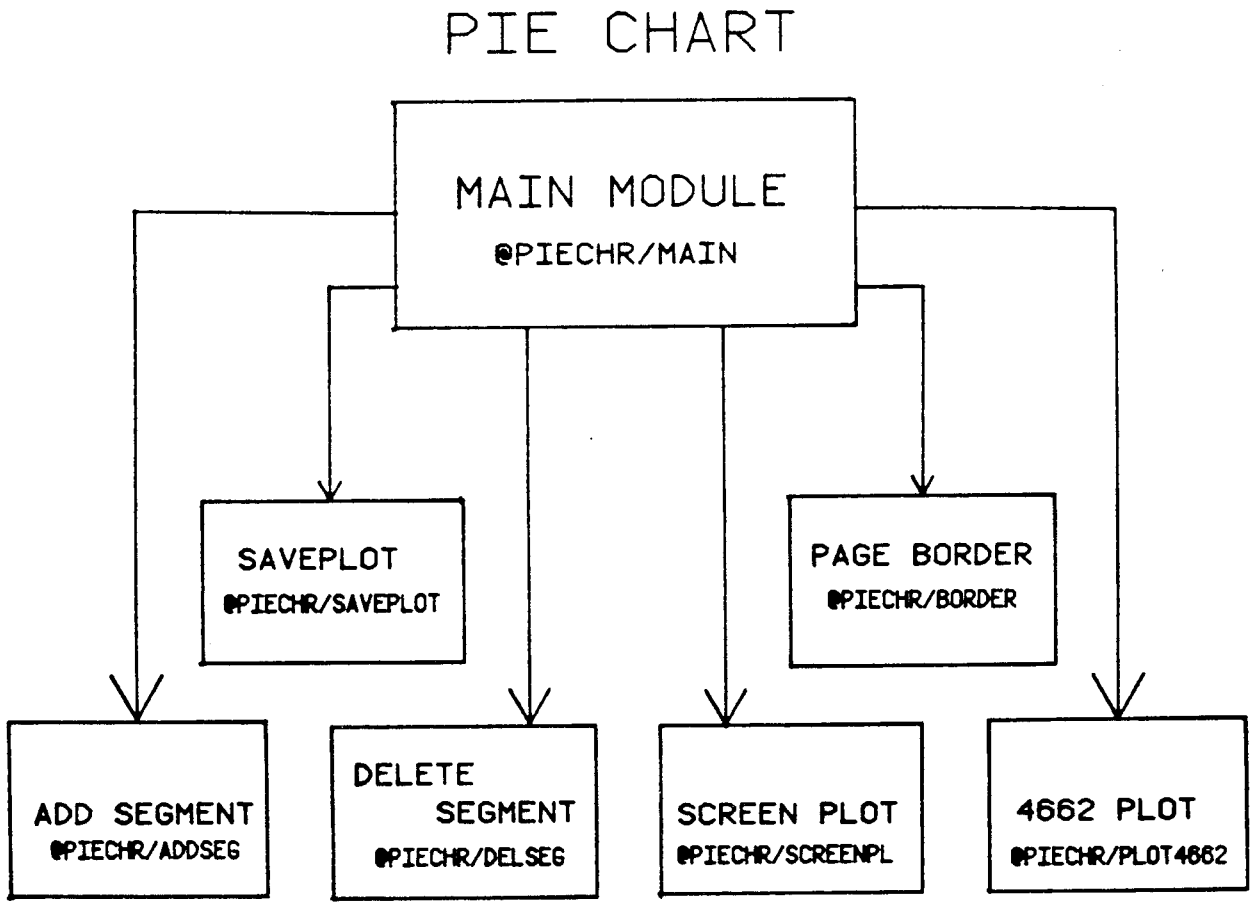
The program prompts the user for the following:

1. Lower left corner of plotting area
2. Upper right corner of plotting area
3. Width and height of characters in the title
4. Positioning of the title
5. Segment labels printing option
6. Symbol to be printed with segment value
7. Segment value to be printed, raw user input or computed percentage
8. Width and height of characters in segment labels
9. Segment separation option
10. Percentage of segment separation if separation is required
11. Position of center of pie
12. Position of top of pie

After all the segments of the pie chart have been drawn, the user is prompted by messages on the screen to position the plotting pen for segment labels and pointers where necessary. To assist the user the plotting pen is moved under program control to the midpoint of the arc of the segment for which a label is to be printed. For all labels to be printed on the right side of the pie chart, the program schecks whether the full text will fit the available space between the pen position and the right margin of the plotting area. If adjustment of pen position is required, this is made by the program to either print the entire label on one line or on multiple lines. The user's responses to programmed prompts are in the Yes/No form. For all labels to be printed on the left side of the pie chart, the pen, after it has been positioned by the user, is moved to indicate where the segment label will end if printed on one line. The options available to the user are:

1. The label should be printed on one line
2. The label should be printed on multiple lines
3. The plotting pen is to be repositioned for another check

TITLE	ABSTRACT NO:
ENHANCED PIE CHART	51/07-9539/0



TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

PROGRAM DATA FILES

Data Structure and Storage

All user supplied and system generated data is stored in Binary format on either tape or disk. Tape files must have been previously marked. Disk files, if not already in existance, will be created and assigned a user supplied name prefixed with "@PIECHR/". Where the data had been previously stored and then read into memory, the number of the tape file or the name of the disk file from which the data was read will be displayed if the same storage media is selected by the user.

Program Files

All of the following modules are stored on disk.

- @PIECHR/MAIN - This is the control module of the pie chart program. It defines the User-Definable Keys, collects chart data, displays the chart data and controls the overlaying of subroutines.
- @PIECHR/ADDSEG - The function of this subroutine is to add segments to an existing pie chart.
- @PIECHR/DELSEG - This subroutine deletes specified segments from and existing pie chart.
- @PIECHR/SCREENPL - This subroutine draws the pie chart on the 4050 screen.
- @PIECHR/PLOT4662 - This subroutine draws the pie chart on the 4662 Plotter.
- @PIECHR/SAVEPLOT - The function of this subroutine is to write the chart data currently in memory on a user specified media - tape or disk.
- @PIECHR/BORDER - This subroutine draws a page border around the defined plotting area on the 4662 Plotter.

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

PROGRAM LOADING

The main program and subroutines are stored on disk. The main program may be loaded under the control of another program or with the command:

```
OLD "@PIECHR/MAIN"
```

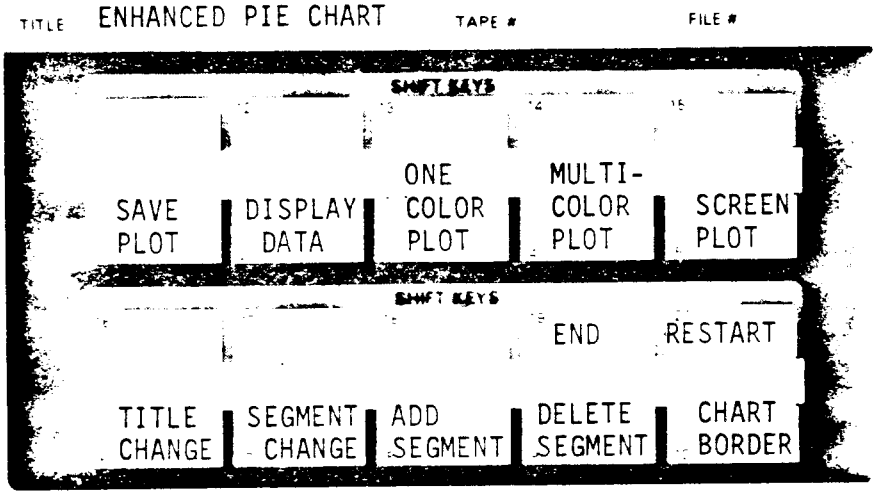
The required subroutines are loaded from disk by the main program with the command:

```
APPEND "@PIECHR/Subroutine";line number
```

PROGRAM EXECUTION

The main program "@PIECHR/MAIN" resides in memory throughout execution. All subroutines are requested through the User-Definable Keys. Each subroutine is assigned a number and when its corresponding User-Definable Key is pressed, a variable used as a switch is set to the subroutine's assigned number. Control is then passed to an overlay control module which compares the switch indicating the required subroutine to another switch indicating the subroutine last loaded into memory. If the required subroutine is already in memory, control is passed directly to the subroutine. If it is not in memory, an APPEND command is executed to load the subroutine and the last-subroutine-loaded switch is reset to the new value corresponding to the loaded subroutine.

TITLE	ABSTRACT NUMBER
ENHANCED PIE CHART	51/07-9539/0



UDK#	FUNCTION
1	Allows the user to save a chart description file on either tape or disk
2	Displays in tabular form on the screen the description of the chart currently sotred in memory
3	Plots the pie chart in one color on the 4662 Plotter
4	Plots the pie chart on the 4662 Plotter, prompting the user to change pen colors at predetermined points
5	Draws the pie chart on the 4050 screen
6	Allows the user to change individual title lines as needed
7	Allows the user to change segment labels and values as needed
8	Allows segments to be added to the chart as needed by entering the required data
9	Allows individual segments of the chart to be deleted

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

UDK #	FUNCTION
10	Draws page borders
19	Terminate program execution
20	Restarts the program at the beginning

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

WELCOME TO THE PIECHART PROGRAM

IN THE VERY UNLIKELY EVENT THAT YOU NEED ASSISTANCE
CONTACT : MALLORY GREEN
OR
ISAAC ARTHUR
HOUSING AND URBAN DEVELOPMENT
PHONE - 202-755-6800

G O O D L U C K

PRESS 'RETURN' TO CONTINUE

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
ENTER ONE OF THE FOLLOWING DEPENDING ON REQUIRED FUNCTION
 N - START A NEW PIE CHART
 T - USE AN EXISTING CHART ON TAPE
 D - USE AN EXISTING CHART ON DISK
ENTER SELECTED FUNCTION



PURPOSE : THIS PROGRAM IS DESIGNED TO INTERACTIVELY
    CREATE A PIE-CHART

FEATURES :

    1   ACCEPTS MAXIMUM OF 3 TITLE LINES
        MAXIMUM OF 64 CHARACTERS PER TITLE LINE
    2   ACCEPTS A MAXIMUM OF 10 SEGMENTS
    3   SEGMENT TEXT LENGTH MAXIMUM IS 25 CHARACTERS
    4   PRINT SEGMENT VALUE EITHER AS A PERCENTAGE
        OR RAW INPUT VALUE AS SUPPLIED BY USER
    5   SEGMENT SEPARATION OPTIONS ARE :
        A. ALL SEGMENTS SEPARATED FROM EACH OTHER
        B. ONE SEGMENT SEPARATED FROM THE REST
        C. NO SEPARATION OF SEGMENTS
    6   POSITIONING OF PEN OR CURSOR FOR SEGMENT TEXT
    7   POSITIONNG OF PEN OR CURSOR FOR SEGMENT/TEXT POINTER
        IF TEXT IS PRINTED OUTSIDE SEGMENT
    8   EACH SEGMENT MAY BE PLOTTED IN A DIFFERENT COLOR
        WHEN PIE CHART IS DRAWN ON THE 4662 PLOTTER


              PRESS 'RETURN' TO CONTINUE
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
NEW PIE CHART ROUTINE
ENTER TITLE 1
REGIONAL SHARE OF TOTAL FEDERAL AID - FY75
ENTER TITLE 2
FROM PRESIDENT'S NATIONAL URBAN POLICY REPORT
ENTER TITLE 3
PAGE 107

HOW MANY SEGMENTS IN THE PIE ? 4
ENTER SEGMENT TEXT AND VALUES
PRESS 'RETURN' AFTER EACH ENTRY

                SEGMENT #          VALUE          TEXT
                    1               26.5          NORTHEAST
                    2               23.4          MIDWEST
                    3               30.9          SOUTH
                    4               19.2          WEST

ROUTINE IS FINISHED . . . CONTINUE EXECUTION VIA USER KEYS
```

UDK #2 - DISPLAY DATA

```
TITLES :

  1 REGIONAL SHARE OF TOTAL FEDERAL AID - FY75
  2 FROM PRESIDENT'S NATIONAL URBAN POLICY REPORT
  3 PAGE 107

SEGMENT TEXT AND VALUES  :

  1  NORTHEAST                    26.5
  2  MIDWEST                      23.4
  3  SOUTH                        30.9
  4  WEST                         19.2


ROUTINE IS FINISHED . . . CONTINUE EXECUTION VIA USER KEYS
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

UDK #1 - SAVE PLOT

```
DO YOU WANT CHART SAVED ON DISK OR TAPE? (D/T) :D

USER NAME AND FILE ARE LIMITED TO 10 CHARACTERS
ENTER (User name/File name) ? IKES/PIETEST


   PLOT DATA SAVED . . .

ROUTINE IS FINISHED . . . CONTINUE EXECUTION VIA USER KEYS
```

UDK #3 - ONE COLOR PLOT

```
YOU HAVE SELECTED ONE-COLOR PLOT


MOVE PEN TO LOWER LEFT CORNER OF YOUR PLOTTING PAPER
  PRESS `RETURN' KEY WHEN  READY

MOVE PEN TO UPPER RIGHT CORNER OF YOUR PLOTTING PAPER
  PRESS `RETURN' KEY  WHEN READY
PLOTTING PEN WILL BE MOVED TO TRACE THE BOUNDARY OF
YOUR PLOTTING AREA
   WATCH THE PEN NOW
 DO YOU WISH TO CHANGE PLOTTING AREA (Y/N) :N

SPECIFY WIDTH AND HEIGHT OF LETTERS FOR THE TITLE
ENTER WIDTH (STANDARD IS 1.75) 2
ENTER HEIGHT (STANDARD IS 2.85) 3.1
DO YOU WANT TITLES CENTERED (Y/N) ? Y
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
SEGMENT TEXT AND VALUES MAY BE :
 1  --  PRINTED WITHIN THE SEGMENT
 2  --  PRINTED OUTSIDE THE SEGMENT WITH A POINTER
 3  --  NOT PRINTED
SELECT OPTION - 1, 2, OR 3 ? 2

ENTER SYMBOL ($,%,...) FOR VALUES: %

USE VALUES INPUT (V), OR COMPUTED PERCENTAGE (P)? V

SPECIFY WIDTH AND HEIGHT OF LETTERS FOR PIE SEGMENT TEXT
ENTER WIDTH (STANDARD IS 1.75)
STANDARD WIDTH WILL BE USED
ENTER HEIGHT (STANDARD IS 2.85)
STANDARD HEIGHT WILL BE USED

ALL SEGMENTS MAY BE SEPARATED OR A SINGLE SEGMENT MAY
SEPARATED FROM THE REST

SEGMENT SEPARATION OPTIONS ARE :-
  0  --  NO SEPARATION
  n  --  VALID SEGMENT NUMBER
  A  --  ALL SEGEMNT TO BE SEPARATED

ENTER SEGMENT SEPARATION OPTION (0/n/A) 0

POSITION PEN AT INTENDED CENTER OF YOUR PIE
   PRESS `RETURN' TO CONTINUE

POSITION PEN AT INTENDED TOP OF YOUR PIE
     PRESS `RETURN' TO CONTINUE

 PLOTTING PEN WILL BE MOVED TO TRACE THE BOUNDARY
 OF THE AREA WITHIN WHICH THE PIECHART WILL BE DRAWN
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	5 1/07-9539/0

```
 WATCH THE PLOTTER PEN NOW
DO YOU WISH TO CHANGE PIECHART AREA? (Y/N) :Y

POSITION PEN AT INTENDED CENTER OF YOUR PIE
   PRESS `RETURN' TO CONTINUE

POSITION PEN AT INTENDED TOP OF YOUR PIE
     PRESS `RETURN' TO CONTINUE

 PLOTTING PEN WILL BE MOVED TO TRACE THE BOUNDARY
 OF THE AREA WITHIN WHICH THE PIECHART WILL BE DRAWN
 WATCH THE PLOTTER PEN NOW
DO YOU WISH TO CHANGE PIECHART AREA? (Y/N) :N
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
SEGMENT TEXT AND VALUES WILL NOW BE PRINTED
HAVING COME THIS FAR I AM SURE YOU DON'T WANT
TO MESS UP YOUR CHART SO PLEASE FOLLOW INSTRUCTIONS
ON THE SCREEN CLOSELY
SEGMENT LABELS WILL BE PRINTED OUTSIDE THE SEGMENTS


SEGMENTS ARE NUMBERED FROM 1 TO 4 STARTING FROM
THE 12 O'CLOCK POSITION AND COUNTING CLOCKWISE

PEN POSITION INDICATES MID-POINT OF ARC FOR
SEGMENT 1. USE IT AS A GUIDE TO POSITION
PEN FOR SEGMENT LABEL 15 CHARACTERS LONG
              PRESS `RETURN' WHEN READY
POSITION PEN FOR LABEL===>SEGMENT POINTER
      PRESS `RETURN' WHEN READY

PEN POSITION INDICATES MID-POINT OF ARC FOR
SEGMENT 2. USE IT AS A GUIDE TO POSITION
PEN FOR SEGMENT LABEL 13 CHARACTERS LONG
              PRESS `RETURN' WHEN READY
POSITION PEN FOR LABEL===>SEGMENT POINTER
      PRESS `RETURN' WHEN READY

PEN POSITION INDICATES MID-POINT OF ARC FOR
SEGMENT 3. USE IT AS A GUIDE TO POSITION
PEN FOR SEGMENT LABEL 11 CHARACTERS LONG
              PRESS `RETURN' WHEN READY
PEN POSITION INDICATES WHERE SEGMENT LABEL WILL END
WHEN PRINTED ON 1 LINE.
     ENTER :      `P' TO PRINT
                  `M' TO PRINT ON MULTIPLE LINES
                  `R' TO REPOSITION PEN MANUALLY P
POSITION PEN FOR LABEL===>SEGMENT POINTER
      PRESS `RETURN' WHEN READY
```

TITLE	ABSTRACT NO:
ENHANCED PIE CHART	51/07-9539/0

```
PEN POSITION INDICATES MID-POINT OF ARC FOR
SEGMENT 4. USE IT AS A GUIDE TO POSITION
PEN FOR SEGMENT LABEL 10 CHARACTERS LONG
              PRESS `RETURN' WHEN READY
PEN POSITION INDICATES WHERE SEGMENT LABEL WILL END
WHEN PRINTED ON 1 LINE.
     ENTER :       `P' TO PRINT
                   `M' TO PRINT ON MULTIPLE LINES
                   `R' TO REPOSITION PEN MANUALLY R
REPOSITION PEN AND PRESS `RETURN'
PEN POSITION INDICATES WHERE SEGMENT LABEL WILL END
WHEN PRINTED ON 1 LINE.
     ENTER :       `P' TO PRINT
                   `M' TO PRINT ON MULTIPLE LINES
                   `R' TO REPOSITION PEN MANUALLY P
POSITION PEN FOR LABEL===>SEGMENT POINTER
      PRESS `RETURN' WHEN READY

ROUTINE IS FINISHED . . . CONTINUE EXECUTION VIA USER KEYS
```

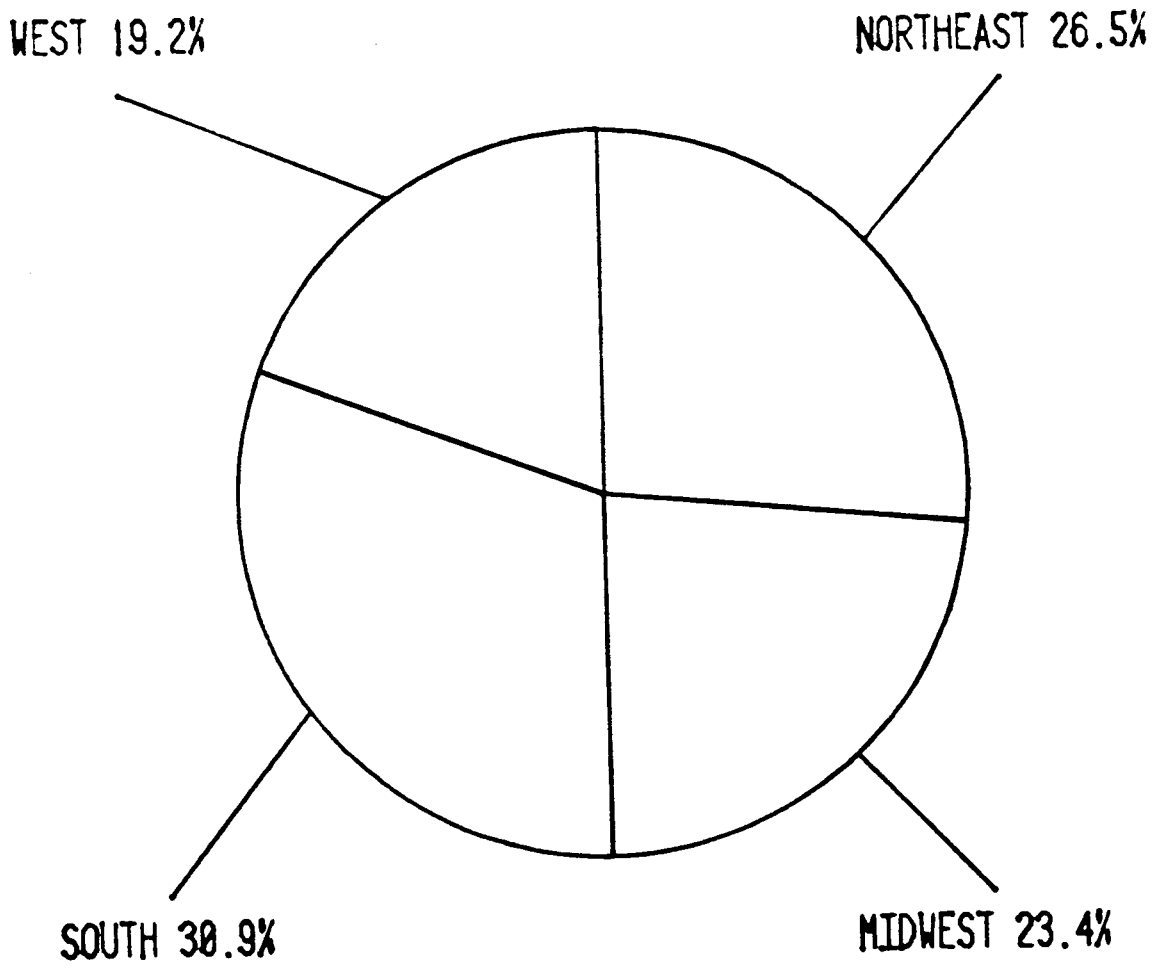
TITLE

ENHANCED PIE CHART

ABSTRACT NO

51/07-9539/0

REGIONAL SHARE OF TOTAL FEDERAL AID - FY75
FROM PRESIDENT'S NATIONAL URBAN POLICY REPORT
PAGE 107



TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

UDK #7 - SEGMENT CHANGE

SEGMENT #	VALUE	TEXT
1	26.5	NORTHEAST
2	23.4	MIDWEST
3	30.9	SOUTH
4	19.2	WEST

```
ENTER NUMBER OF SEGMENT TO BE CHANGED OR '0' TO EXIT
WHICH SEGMENT #  : 1
ENTER NEW TEXT :
ENTER NEW VALUE : 30.5
NEXT SEGMENT TO BE CHANGED ? 2
ENTER NEW TEXT : 26.4
ENTER NEW VALUE : 26.4
NEXT SEGMENT TO BE CHANGED ? 2
ENTER NEW TEXT : MIDWEST
ENTER NEW VALUE :
NEXT SEGMENT TO BE CHANGED ? 3
ENTER NEW TEXT :
ENTER NEW VALUE : 20.9
NEXT SEGMENT TO BE CHANGED ? 4
ENTER NEW TEXT : 22.2
ENTER NEW VALUE : 22.2
NEXT SEGMENT TO BE CHANGED ? 4
ENTER NEW TEXT : WEST
ENTER NEW VALUE :
NEXT SEGMENT TO BE CHANGED ? 0

ROUTINE IS FINISHED . . . CONTINUE EXECUTION VIA USER KEYS
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

UDK #5 - SCREEN PLOT

```
SEGMENT TEXT AND VALUES MAY BE:
 1  --  PRINTED WITHIN THE SEGMENT
 2  --  OUTSIDE THE SEGMENT
 3  --  NOT PRINTED
SELECT OPTION -- 1, 2, OR 3 ? 2

ENTER SYMBOL ($,%,...) FOR VALUES: %

USE INPUT VALUES (V), OR COMPUTED PERCENTAGE (P)? P

ENTER % OF WINDOW ALLOTTED TO CIRCLE (0-100%): 90

 ALL SEGMENTS MAY BE SEPARATED OR A SINGLE
SEGEMNT MAY BE SEPARATED FROM THE REST

SEGMENT SEPARATION OPTIONS ARE :-
 0 -- NO SEPARATION
 n -- VALID SEGMENT NUMBER
 A -- ALL SEGMENTS TO BE SEPARATED

ENTER SEGMENT SEPARATION OPTION (0/n/A) 0
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

THE ATTENTION BELL WILL RING AFTER EACH SEGMENT HAS BEEN
DRAWN. POSITION THE CURSOR, USING THE JOYSTICK, FOR
THE SEGMENT TEXT AND VALUE TO BE PRINTED.
PRESS THE 'RETURN' KEY AFTER POSITIONING THE CURSOR
IF SEGMENT TEXT IS PRINTED OUTSIDE THE SEGMENT
(OPTION 2), THE ATTENTION BELL WILL RING FOR
CURSOR TO BE POSITIONED FOR SEGMENT/TEXT POINTER
PRESS 'RETURN' KEY AFTER POSITIONING CURSOR
THE TABLE BELOW SHOWS THE LENGTH OF EACH SEGMENT TEXT.
USE IT AS A GUIDE IN POSITIONING THE CURSOR

SEGMENT LABELS WILL BE PRINTED OUTSIDE THE SEGMENTS
SEGMENT # 1 TEXT LENGTH = 10
SEGMENT # 2 TEXT LENGTH = 8
SEGMENT # 3 TEXT LENGTH = 6
SEGMENT # 4 TEXT LENGTH = 5

YOU MAY MAKE A HARD COPY OF ABOVE INSTRUCTIONS FOR
REFFERENCE. PLEASE NOTE NO FURTHER INSTRUCTIONS WILL
APPEAR ON THE SCREEN ONCE THE PLOTTING HAS BEGUN

PRESS 'RETURN' WHEN READY TO CONTINUE

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

DATA STORAGE

MAIN MODULE - @PIECHR/MAIN

Variable	Used to Store	Type
A$	Segment labels	String
D	Segment values	Array (10)
D$	Title lines entered by user	String
F$	Title line 1	String
G$	Title line 2	String
H$	Title line 3	String
I	Program index	Simple
I1	Program index	Simple
I$	Source of chart data (D/T/N)	String
J$	Chart data disk file name	String
N	Number of segments	Simple
N$	Input of segment values	String
O$	Scratch used as dummy response	String
P	Segment values as percentages of total pie	Array (11)
P1	Scratch	Simple
P2	Scratch	Simple
S	Sum of segment values	Simple
S1	Number assigned to required subroutine	Simple
S2	Number of subroutine last read from disk	Simple
T$	Segment labels input by user	String
U$	Indicator for multi-color plot	String
Y$	Segment values	String
Z8	Tape file number for chart data	Simple
Z9	Scratch	Simple
Z$	Disk file status message	String

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

SEGMENT ADDITION - @PIECHR/ADDSEG

Variable	Used to Store	Type
A1	Number of segments to be added	Simple
A2	New total of segments in the pie	Simple
A$	All segment labels	String
D	Segment values	Array (10)
I	Program index	Simple
I1	Program index	Simple
N	Original number of segments	Simple
N$	Segment value	String
P1	Scratch	Simple
S	Sum of all segment values	Simple
T$	Scratch	String
Y$	All Segment values	String

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

SEGMENT DELETION - @PIECHR/DELSEG

Variable	Used to Store	Type
A1	Number of segments to be deleted	Simple
A$	All segment labels	String
D	Segment values	Array (10)
D1	Total number of segments deleted	Simple
I	Index	Simple
I1	Index	Simple
I2	Index	Simple
N	Number of segments	Simple
P1	Scratch	Simple
P2	Scratch	Simple
P3	Scratch	Simple
P4	Scratch	Simple
Q$	Scratch	String
S	Sum of all segment values	Simple
Y$	All segment values	String

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

SCREEN PLOT - @PIECHR/SCREENPL

Variable	Used to Store	Type
A$	All segment labels	String
B1	Beginning angle of a segment's arc	Simple
B2	Ending angle of a segment's arc	Simple
B3	Angle of the midpoint of a segment's arc	Simple
B$	A segment's label	String
D	Segment values	Array (10)
D$	Scratch	String
E	Beginning angle of a segment's arc	Simple
E1	Segment separation value	Simple
F$	Title line 1	String
G$	Title line 2	String
H	Character height scale factor	Simple
H$	Title line 3	String
I	Program index	Simple
I1	Program index	Simple
I6	Scratch	Simple
I$	Scratch used for user response	Simple
J	Ending angle for a segment's arc	Simple
L	Scratch	Simple
N	Number of segments	Simple
N1	Segment to be separated	Simple
N$	Segment values	String
O$	Scratch used for dummy response	String
P	Segment values as percentages of total pie	Array (11)
P1	Scratch	Simple
P2	Scratch	Simple
P6	Scratch	Simple
P7	Scratch	Simple
P9	Segment label printing option	Simple
P$	Scratch	String

TITLE

ENHANCED PIE CHART

ABSTRACT NO

51/07-9539/0

Variable	Used to Store	Type
QC	Scratch	Simple
S	Sum of all segment values	Simple
S$	Symbol to be printed with segment values	String
VS	Segment value (raw or computed %) to be printed option	String
W	User units for plotting window	Simple
W1	Character width scale factor	Simple
W2	Percentage of window alloted to pie chart	Simple
W3	Scratch	Simple
X	Adjustment of window on X axis for segment separation	Simple
X1	X axis coordinate of mid-point of segment arc	Simple
X2	X axis coordinate of beginning os segment pointer	Simple
X3	Beginning of title print position	Simple
Y	Adjustment of window on Y axis for segment separation	Simple
Y1	Y axis coordinate of mid-point of segment arc	Simple
Y2	Y axis coordinate of beginning of segment pointer	Simple
Y$	All segment values	String
Z	Device number on GPIB	

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

4662 PLOT - @PIECHR/PLOT4662

Variable	Used to Store	Type
A1	X coordinate of end of first segment's arc	Simple
A2	Y coordinate of end of first segment's arc	Simple
A$	Segment labels	String
B1	Beginning angle of segment's arc	Simple
B2	Ending angle of segment's arc	Simple
B3	Angle of midpoint of segment's arc	Simple
B7	X axis coordinate of lower left corner of plotting area	Simple
B8	Y axis coordinate of lower left corner of plotting area	Simple
B$	A segment's label	String
C	Scratch	Simple
C1	Scratch	Simple
C$	Underlining characters	String
D	Segment values	Array (10)
E	Beginning angle of a segment's arc	Simple
E1	Percentage of segment spearation	Simple
E7	Adjusted X axis beginning	Simple
E8	Adjusted X axis ending	Simple
F$	Title line 1	String
G$	Title line 2	String
H	Character height scaling factor	Simple
H1	X axis coordinate of midpoint of segment arc	Simple
H9	Adjusted character height scaling factor	Simple
H$	Title line 3	String
I$	Scratch used for Y/N responses	String
I	Index	Simple
I1	Index	Simple
I3	Index	Simple
J	Ending angle of segment arc	Simple

TITLE

ENHANCED PIE CHART

ABSTRACT NO

51/07-9539/0

Variable	Used to Store	Type
L	Scratch	Simple
L1	Scratch	Simple
L$	Segment label printing option	String
M1	Number of decimal places to which segment value should be computed and printed	Simple
M$	User input for number of decimal places to which segment value should be printed	String
N	Number of segments	Simple
N1	Segment to be separated	Simple
N9	Rounding up of segment values	Simple
N$	Segment value	String
O$	Scratch	String
P	Computed segment percentages of total pie	Array (11)
P1	Scratch	Simple
P2	Scratch	Simple
P9	Segment label printing option	Simple
R	Radius of pie	Simple
R1	Adjustment of X axis range of viewport	Simple
R$	Scratch	String
S	Sum of segment values	Simple
S$	Symbol to be printed with segment value	String
T$	Scratch	String
U$	Multi-color plot indicator	String
V1	Y axis coordinate of midpoint of segment arc	Simple
V$	Segment value to be printed option	String
W	User units for window	Simple
W1	Character width scaling factor	Simple
W2	Percentage of plotting area for chart (always 100)	Simple
W3	Angle of shift for segment separation	Simple
W$	Character width scaling factor input by user	String

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

Variable	Used to Store	Type
X	X axis adjustment of window for segment separation	Simple
XØ	X axis coordinate of center of pie	Simple
X1	X axis coordinate of lower left corner of plotting paper	Simple
X2	X axis coordinate of upper right corner of plotting paper	Simple
X3	Beginning print position for centered titles	Simple
X4	GDU of plotting area on the X axis	Simple
X6	X axis coordinate of top of pie	Simple
X7	Window adjustment factor for segment separation	Simple
X8	Scratch	Simple
X9	Beginning of segment label print position	Simple
X$	Scratch	String
Y	Y axis adjustment of window for segment separation	Simple
YØ	Y axis coordinate of center of pie	Simple
Y1	Y axis coordinate of lower left corner of plotting area	Simple
Y2	Y axis coordinate of upper right corner of plotting area	Simple
Y6	Y axis coordinate of top of pie	Simple
Y7	Window adjustment factor for segment separation	Simple
Y9	Beginning of segment label print position	Simple
Y$	All segment values	String
Z	Device number of GPIB	Simple
Z1	Scratch	Simple
Z2	Scratch	Simple
Z$	Scratch	String

TITLE	ABSTRACT NO
Enhanced Pie Chart	51/07-9539/0

SAVE PLOT - @PIECHR/SAVEPLOT

Variable	Used to Store	Type
A$	All segment labels	String
D	All segment values	Array (10)
F$	Title line 1	String
G$	Title line 2	String
H$	Title line 3	String
J$	User's disk file name for chart data	String
M$	Storage media for file	String
N	Number of segments	Simple
S	Sum of all segment values	Simple
Y$	All segment values	String
Z$	Scratch	String

TITLE

ENHANCED PIE CHART

ABSTRACT NO

51/07-9539/0

CHART PAGE BORDER - @PIECHR/BORDER

Variable	Used to Store	Type
I$	Scratch	String
O$	Scratch	String
X1	X axis coordinate of lower left corner of plotting area	Simple
X2	X axis coordinate of upper right corner of plotting area	Simple
X3	User display units of window on X axis	Simple
Y1	Y axis coordinate of lower left corner of plotting area	Simple
Y2	Y axis coordinate of upper right corner of plotting area	Simple
Y3	User display units of window on Y axis	Simple
Z	Device number on GPIB	Simple

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1 REMARK *PIECHART PROGRAM MAIN MODULE @PIECHR/MAIN 51/07-9539/0

2 GOSUB 100
3 GO TO 90

4 REMARK *** SAVE PLOT DATA

5 S1=1
6 GOSUB 2530
7 GO TO 90

8 REM * DISPLAY PLOT DATA

9 GOSUB 2360
10 GO TO 90

12 REM *** ONE COLOR PLOT

13 U$="U"
14 GO TO 93

16 REM *** MULTI-COLOR PLOT

17 U$=" "
18 GO TO 93

20 REM *** PLOT TO SCREEN

21 S1=5
22 GOSUB 2530
23 GO TO 90

24 REMARK *** CHART TITLE CHANGES

25 GOSUB 1500
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

```
_6 GO TO 90

28 REMARK *** SEGMENT TEXT AND VALUE CHANGES

29 GOSUB 1850
30 GO TO 90

32 REMARK *** ADD A NEW SEGMENT

33 S1=2
34 GOSUB 2530
35 GO TO 90

36 REMARK *** DELETE A SEGMENT

37 S1=3
38 GOSUB 2530
39 GO TO 90

40 REMARK *** DRAW PAGE BORDER

41 S1=4
42 GOSUB 2530
43 GO TO 90

76 REMARK *** END OF RUN

77 PRINT "LJJJJIIHHHHEND  OF  RUNGGG"
78 END

80 REMARK *** RESTART

81 GOSUB 220

90 REMARK
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
91 PRINT "JROUTINE IS FINISHED . . . CONTINUE EXECUTION VIA USER KEYSG"
92 RETURN

93 REMARK *** SET UP TO BRING IN PLOT 4662 MODULE

94 S1=6
95 GOSUB 2530
96 GO TO 90

100 REMARK *** INTRODUCTION

110 SET KEY
120 PRINT "LJ         WELCOME TO THE PIECHART PROGRAM"
130 PRINT "JJ     IN THE VERY UNLIKELY EVENT THAT YOU NEED ASSISTANCE"
140 PRINT "              CONTACT : MALLORY GREEN"
150 PRINT "                                OR "
160 PRINT "                             ISAAC ARTHUR"
170 PRINT "                             HOUSING AND URBAN DEVELOPMENT"
180 PRINT "                             PHONE - 202-755-6800"
190 PRINT "JJ                           G O O D  L U C K"
200 PRINT "JJIPRESS  'RETURN'  TO CONTINUE ";
210 INPUT O$
220 DELETE A$,D,P,Z$,N$,Y$
230 DIM A$(10*25),D(10),P(11),Z$(25)
240 SET DEGREES
250 S1=0
260 S2=0
270 PRINT "LENTER ONE OF THE FOLLOWING DEPENDING ON REQUIRED FUNCTION"
280 PRINT " N - START A NEW PIE CHART"
290 PRINT " T - USE AN EXISTING CHART ON TAPE"
300 PRINT " D - USE AN EXISTING CHART ON DISK"
310 PRINT "ENTER SELECTED FUNCTION ";
320 INPUT I$
330 GO TO 1*(I$="N")+2*(I$="T")+3*(I$="D") OF 360,1290,1390
340 PRINT "GJINVALID FUNCTION SELECTED -- RE-ENTER ";
350 GO TO 320
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
360 PRINT "L PURPOSE : THIS PROGRAM IS DESIGNED TO INTERACTIVELY"
370 PRINT "     CREATE A PIE-CHART "
380 PRINT "JFEATURES :J"
390 PRINT "    1   ACCEPTS MAXIMUM OF 3 TITLE LINES"
400 PRINT "        MAXIMUM OF 64 CHARACTERS PER TITLE LINE"
410 PRINT "    2   ACCEPTS A MAXIMUM OF 10 SEGMENTS              "
420 PRINT "    3   SEGMENT TEXT LENGTH MAXIMUM IS 25 CHARACTERS"
430 PRINT "    4   PRINT SEGMENT VALUE EITHER AS A PERCENTAGE"
440 PRINT "        OR RAW INPUT VALUE AS SUPPLIED BY USER"
450 PRINT "    5   SEGMENT SEPARATION OPTIONS ARE :              "
460 PRINT "        A. ALL SEGMENTS SEPARATED FROM EACH OTHER"
470 PRINT "        B. ONE SEGMENT SEPARATED FROM THE REST "
480 PRINT "        C. NO SEPARATION OF SEGMENTS"
490 PRINT "    6   POSITIONING OF PEN OR CURSOR FOR SEGMENT TEXT"
500 PRINT "    7   POSITIONNG OF PEN OR CURSOR FOR SEGMENT/TEXT POINTER"
510 PRINT "        IF TEXT IS PRINTED OUTSIDE SEGMENT"
520 PRINT "    8   EACH SEGMENT MAY BE PLOTTED IN A DIFFERENT COLOR"
530 PRINT "        WHEN PIE CHART IS DRAWN ON THE 4662 PLOTTER"
540 PRINT "JJIPRESS  'RETURN'  TO CONTINUE";
550 INPUT O$
560 PRINT "LNEW PIE CHART ROUTINE"
570 FOR I=1 TO 3
580     PRINT "ENTER TITLE ";I
590     INPUT D$
600     IF LEN(D$)<65 THEN 630
610     PRINT "GTITLE EXCEEDS 64 CHARACTERS"
620     GO TO 590
630     IF I>2 THEN 690
640     IF I=2 THEN 670
650     F$=D$
660     GO TO 700
670     G$=D$
680     GO TO 700
690     H$=D$
700 NEXT I
710 Z8=0
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
 20 J$=""
730 Y$=""
740 PRINT "JHOW MANY SEGMENTS IN THE PIE ? ";
750 INPUT N
760 IF N<=10 THEN 800
770 PRINT "GMORE THAN 10 SEGMENTS WILL PRODUCE A CROWDED CHART"
780 PRINT "GMAXIMUM OF TEN SEGMENTS WILL BE DRAWN "
790 N=10
800 IF N>0 THEN 830
810 PRINT "GYOU CANNOT DRAW A PIE CHART WITH 0 SEGMENTS "
820 GO TO 740
830 A$=""
840 F$=F$&" "
850 G$=G$&" "
860 H$=H$&" "
870 S=0
880 FOR I=1 TO 10
890     D(I)=0
900 NEXT I
910 FOR I=1 TO 11
920     P(I)=0
930 NEXT I
940 PRINT "ENTER SEGMENT TEXT AND VALUES"
950 PRINT "PRESS 'RETURN' AFTER EACH ENTRY"
960 PRINT "JIHHHHHSEGMENT #IHHVALUEIHHHHTEXT"
970 FOR I=1 TO N
980     PRINT "I";I;"I";
990     INPUT N$
1000     P1=POS(N$,",",1)
1010     IF P1=0 THEN 1040
1020     N$=REP("",P1,1)
1030     GO TO 1000
1040     PRINT "KIIIHHHHHHH";
1050     INPUT T$
1060     IF LEN(T$)<26 THEN 1100
1070     PRINT "GTEXT LONGER THAN 25 CHARACTERS - RE-ENTER"
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1080      PRINT "III8HHHHH";
1090      GO TO 1050
1100      A$=A$&T$
1110      IF LEN(T$)=25 THEN 1150
1120      FOR I1=1 TO 25-LEN(T$)
1130          A$=A$&" "
1140      NEXT I1
1150      D(I)=VAL(N$)
1160      IF D(I)>0 THEN 1220
1170      PRINT "G A SEGMENT MAY NOT HAVE A VALUE OF 0 OR LESS";
1180      PRINT " RE-ENTER"
1190      PRINT "II";
1200      INPUT N$
1210      GO TO 1150
1220      S=S+D(I)
1230      N$=STR(D(I))
1240      N$=SEG(N$,2,LEN(N$)-1)
1250      Y$=Y$&N$
1260      Y$=Y$&" "
1270 NEXT I
1280 GO TO 1490

1290 REMARK *** READ PLOT FORM TAPE

1300 PRINT "GINSERT TAPE & ENTER FILE NUMBER WHERE CHART IS STORED :";
1310 INPUT Z8
1320 FIND Z8
1330 ON EOF (0) THEN 1380
1340 READ @33:A$,F$,G$,H$,D,N,S,Y$
1350 DIM P(11),Z$(25)
1360 GOSUB 2360
1370 J$=""
1380 GO TO 1490

1390 REMARK *** READ PLOT FROM DISK
```

TITLE	ABSTRACT NO
ENHANCED PEI CHART	51/07-9539/0

```
1400 PRINT "GJENTER (User name/File name) ? ";
1410 INPUT J$
1420 J$="@PIECHR/"&J$
1421 CALL "FILE",0,J$,Z$
1422 IF Z$<>"" THEN 1430
1423 J$=SEG(J$,9,LEN(J$)-8)
1424 PRINT "JGGGG NO SUCH FILE.................";J$;"GGGGG"
1425 GO TO 1480
1430 OPEN J$;1,"R",N$
1440 READ #1:A$,F$,G$,H$,D,N,S,Y$
1450 DIM P(11),Z$(25)
1460 CLOSE 1
1470 GOSUB 2360
1480 Z8=0
1490 RETURN

1500 REMARK *** TITLE LINES CHANGES

1510 PAGE
1520 PRINT "TITLE 1 :"
1530 PRINT "       ";F$
1540 PRINT "TITLE 2 :"
1550 PRINT "       ";G$
1560 PRINT "TITLE 3 :"
1570 PRINT "       ";H$
1580 PRINT "JENTER NUMBER OF TITLE TO BE CHANGED OR `0' TO EXIT"
1590 PRINT "ENTER NUMBER ";
1600 INPUT Z9
1610 IF Z9<=3 THEN 1640
1620 PRINT "GINVALID ENTRY -- RE-ENTER ";
1630 GO TO 1600
1640 IF Z9<=0 THEN 1840
1650 PRINT "REPLACE TITLE ";Z9;" WITH :"
1660 INPUT D$
1670 IF LEN(D$)<65 THEN 1700
1680 PRINT "GTITLE EXCEEDS 64 CHARACTERS "
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1690 GO TO 1650
1700 IF Z9>2 THEN 1800
1710 IF Z9=2 THEN 1760
1720 F$=D$
1730 IF D$<>"" THEN 1590
1740 F$=" "
1750 GO TO 1590
1760 G$=D$
1770 IF D$<>"" THEN 1590
1780 G$=" "
1790 GO TO 1590
1800 H$=D$
1810 IF D$<>"" THEN 1590
1820 H$=" "
1830 GO TO 1590
1840 RETURN

1850 REMARK *** CHANGE SEGMENT TEXT AND VALUES

1860 PRINT "LJIHHHHHSEGMENT #IVALUEIHHHHTEXTJ"
1870 FOR I=1 TO N
1880     T$=SEG(A$,(I-1)*25+1,25)
1890     PRINT "I";I;"I  ";D(I);"IHHHHHHHH";T$
1900 NEXT I
1910 PRINT "JENTER NUMBER OF SEGMENT TO BE CHANGED OR `0' TO EXIT "
1920 PRINT "WHICH SEGMENT #  : ";
1930 INPUT I
1940 IF I<=N THEN 1970
1950 PRINT "GATTEMPTING TO CHANGE A NON-EXISTENT SEGMENT"
1960 GO TO 1910
1970 IF I<=0 THEN 2310
1980 PRINT "ENTER NEW TEXT : ";
1990 INPUT T$
2000 IF LEN(T$)<26 THEN 2030
2010 PRINT "TEXT EXCEEDS MAX OF 25 CHARACTERS -- RE-ENTER"
2020 GO TO 1990
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
2030 IF LEN(T$)=25 THEN 2080
2040 IF T$="" THEN 2090
2050 FOR I1=1 TO 25-LEN(T$)
2060     T$=T$&" "
2070 NEXT I1
2080 A$=REP(T$,(I-1)*25+1,25)
2090 PRINT "ENTER NEW VALUE : ";
2100 INPUT N$
2110 IF N$="" THEN 2290
2120 P1=POS(N$,",",1)
2130 IF P1=0 THEN 2160
2140 N$=REP("",P1,1)
2150 GO TO 2120
2160 D(I)=VAL(N$)
2170 IF D(I)>0 THEN 2200
2180 PRINT "GA SEGMENT VALUE CANNOT BE ""0"" "
2190 GO TO 2090
2200 P2=1
2210 IF I=1 THEN 2260
2220 FOR I1=1 TO I-1
2230     P1=POS(Y$," ",P2)
2240     P2=P1+1
2250 NEXT I1
2260 P1=POS(Y$," ",P2)
2270 N$=N$&" "
2280 Y$=REP(N$,P2,P1-P2+1)
2290 PRINT "NEXT SEGMENT TO BE CHANGED ? ";
2300 GO TO 1930
2310 S=0
2320 FOR I=1 TO N
2330     S=S+D(I)
2340 NEXT I
2350 RETURN

2360 REMARK *** DISPLAY PLOT DATA
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
2370 PRINT "LTITLES : J"
2380 PRINT "  1 ";F$
2390 PRINT "  2 ";G$
2400 PRINT "  3 ";H$
2410 PRINT "JSEGMENT TEXT AND VALUES  :J "
2420 P2=1
2430 FOR I=1 TO N
2440     P1=POS(Y$," ",P2)
2450     N$=SEG(Y$,P2,P1-P2)
2460     P2=P1+1
2470     T$=SEG(A$,(I-1)*25+1,25)
2480     PRINT "  ";I;"  ";T$;" ";N$
2490 NEXT I
2500 PRINT
2510 PRINT
2520 RETURN

2530 REMARK *** OVERLAY CONTROL MODULE

2540 IF S2=S1 THEN 2660
2550 S2=S1
2560 DELETE 2730,9999
2570 Z9=MEMORY
2580 GO TO S2 OF 2590,2610,2630,2650,2670,2690
2590 APPEND "@PIECHR/SAVEPLOT";2720
2600 GO TO 2700
2610 APPEND "@PIECHR/ADDSEG";2720
2620 GO TO 2700
2630 APPEND "@PIECHR/DELSEG";2720
2640 GO TO 2700
2650 APPEND "@PIECHR/BORDER";2720
2660 GO TO 2700
2670 APPEND "@PIECHR/SCREENPL";2720
2680 GO TO 2700
2690 APPEND "@PIECHR/PLOT4662";2720
2700 GOSUB 2720
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

```
2710 RETURN

2720 REMARK *** TARGET FOR APPEND
2730 REMARK TARGET FOR INITIAL DELETE
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
100 REMARK *** ADD A NEW SEGMENT "@PIECHR/ADDSEG" 51/07-9539/0

110 PAGE
120 PRINT " ENTER NUMBER OF SEGMENTS TO BE ADDED ";
130 INPUT A1
140 IF A1<=0 THEN 480
150 A2=A1+N
160 IF A2<=10 THEN 190
170 PRINT "GYOU MAY ADD MAXIMUM OF ";10-N;" SEGMENTS"
180 GO TO 120
190 PRINT "JIHHHHHSEGMENT #IHHVALUEIHHHHTEXT"
200 FOR I=1 TO A1
210     PRINT "I";I;"I";
220     INPUT N$
230     P1=POS(N$,",",1)
240     IF P1=0 THEN 270
250     N$=REP("",P1,1)
260     GO TO 230
270     IF VAL(N$)>0 THEN 310
280     PRINT "GA SEGMENT MAY NOT HAVE A VALUE OF 0 OR LESS -- RE-ENTER"
290     PRINT "II";
300     GO TO 220
310     PRINT "KIIIHHHHHH";
320     INPUT T$
330     IF LEN(T$)<26 THEN 370
340     PRINT "GTEXT LONGER THAN 25 CHARACTERS -- RE-ENTER"
350     PRINT "IIIHHHHHH";
360     GO TO 320
370     A$=A$&T$
380     IF LEN(T$)=25 THEN 420
390     FOR I1=1 TO 25-LEN(T$)
400         A$=A$&" "
410     NEXT I1
420     D(N+I)=VAL(N$)
430     S=S+D(N+I)
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
440       Y$=Y$&NC
450       Y$=Y$&" "
460 NEXT I
470 N=A2
480 RETURN
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
100 REMARK *** DELETE A SEGMENT "@PIECHR/DELSEG" 51/07-9539/0

110 D1=0
120 PRINT "LENTER NUMBER OF SEGMENT TO BE DELETED"
130 PRINT "       OR '0' TO EXIT ROUTINEJ"
140 PRINT "ENTER NUMBER OF SEGMENT ";
150 INPUT A1
160 IF A1<=0 THEN 410
170 IF A1<=N THEN 200
180 PRINT "GATTEMPTING TO DELETE A NON-EXISTENT SEGMENT"
190 GO TO 140
200 P2=1
210 FOR I1=1 TO A1
220     P1=POS(Y$," ",P2)
230     IF I1=A1 THEN 250
240     P2=P1+1
250 NEXT I1
260 IF A1<>1 THEN 280
270 P2=1
280 Q$=SEG(Y$,P2,P1-P2)
290 GO TO 1*(Q$="****")+2*(Q$<>"****") OF 300,320
300 PRINT "GSEGMENT ALREADY DELETED"
310 GO TO 140
320 IF D1+2<N THEN 380
330 PRINT "GYOU NEED A MINIMUM OF 2 SEGMENTS FOR A MEANINGFUL PIECHART"
340 PRINT "AND YOU NOW HAVE ONLY 2 SEGMENTS"
350 PRINT "GGJNO MORE SEGMENTS WILL BE DELETED UNLESS YOU ADD MORE"
360 PRINT "SEGMENTS FIRST"
370 GO TO 420
380 Y$=REP("****",P2,P1-P2)
390 D1=D1+1
400 GO TO 140
410 P2=1
420 FOR I=1 TO N
430     P1=POS(Y$," ",P2)
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

```
440      Q$=SEG(Y$,P2,P1-P2)
450      IF Q$<>"****" THEN 480
460      Y$=REP("",P2,P1+1-P2)
470      A$=REP("",(I-1)*25+1,25)
480      P2=P1+1
490 NEXT I
500 P2=1
510 S=0
520 FOR I=1 TO N-D1
530      P1=POS(Y$," ",P2)
540      Q$=SEG(Y$,P2,P1-P2)
550      D(I)=VAL(Q$)
560      S=S+D(I)
570      P2=P1+1
580 NEXT I
590 N=N-D1
600 RETURN
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
100 REMARK *** PLOT TO SCREEN "@PIECHR/SCREENPL" 51/07-9539/0

110 PRINT "LJSEGMENT TEXT AND VALUES MAY BE:"
120 PRINT " 1  --  PRINTED WITHIN THE SEGMENT"
130 PRINT " 2  --  OUTSIDE THE SEGMENT"
140 PRINT " 3  --  NOT PRINTED "
150 PRINT "GSELECT OPTION -- 1, 2, OR 3 ? ";
160 INPUT P$
170 P9=VAL(P$)
180 IF P9>0 AND P9<=3 THEN 210
190 PRINT "GINVALID RESPONSE -- RE-ENTER ";
200 GO TO 160
210 IF P9=3 THEN 320
220 PRINT "JENTER SYMBOL ($,%,...) FOR VALUES: ";
230 INPUT S$
240 IF LEN(S$)<=1 THEN 270
250 PRINT "ONLY FIRST CHARACTER OF SYMBOL WILL BE PRINTED"
260 S$=SEG(S$,1,1)
270 PRINT "JUSE INPUT VALUES (V), OR COMPUTED PERCENTAGE (P)? ";
280 INPUT V$
290 IF V$="V" OR V$="P" THEN 320
300 PRINT "GINVALID RESPONSE -- RE-ENTER ";
310 GO TO 280
320 W1=1.75
330 H=2.85
340 PRINT "JENTER % OF WINDOW ALLOTTED TO CIRCLE (0-100%): ";
350 W2=100
360 INPUT Z$
370 IF Z$="" THEN 390
380 W2=VAL(Z$)
390 PRINT "J ALL SEGMENTS MAY BE SEPARATED OR A SINGLE "
400 PRINT "SEGEMNT MAY BE SEPARATED FROM THE REST "
410 PRINT "JSEGMENT SEPARATION OPTIONS ARE :- "
420 PRINT " 0 -- NO SEPARATION "
430 PRINT " n -- VALID SEGMENT NUMBER"
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
440 PRINT " A -- ALL SEGMENTS TO BE SEPARATED "
450 PRINT "JENTER SEGMENT SEPARATION OPTION (0/N/A) ";
460 INPUT Z$
470 IF Z$<>"" THEN 500
480 PRINT "GINVALID RESPONSE "
490 GO TO 450
500 E1=20
510 IF Z$<>"A" THEN 550
520 PRINT "GTO MINIMIZE DISTORTION CAUSED BY SEGMENTS EXPLOSIONS"
530 PRINT "5% MAXIMUM SEPARATION IS RECOMMENDED"
540 GO TO 600
550 N1=VAL(Z$)
560 IF N1=0 THEN 710
570 IF N1<=N THEN 610
580 PRINT "GATTEMPTING TO SEPARATE A NON-EXISTENT SEGMENT"
590 GO TO 450
600 N1=99
610 PRINT "JENTER % SEPARATION -(RECOMMENDED MAX IS 20) ";
620 INPUT E1
630 IF E1>0 THEN 660
640 PRINT " '0' IS NOT A VALID VALUE -- RE-ENTER ";
650 GO TO 620
660 IF E1<=20 THEN 710
670 PRINT "GSEPARATION OF MORE THAN 20% MAY CAUSE LOSS OF"
680 PRINT "          CORRELATION BETWEEN SEGMENTS"
690 FOR I1=1 TO 500
700 NEXT I1
710 IF P9=3 THEN 1010
720 PRINT "LGGGG"
730 PRINT "THE ATTENTION BELL WILL RING AFTER EACH SEGMENT HAS BEEN"
740 PRINT "DRAWN. POSITION THE CURSOR, USING THE JOYSTICK, FOR"
750 PRINT "THE SEGMENT TEXT AND VALUE TO BE PRINTED."
760 PRINT "PRESS THE 'RETURN' KEY AFTER POSITIONING THE CURSOR"
770 PRINT "IF SEGMENT TEXT IS PRINTED OUTSIDE THE SEGMENT"
780 PRINT " (OPTION 2), THE ATTENTION BELL WILL RING FOR "
790 PRINT "CURSOR TO BE POSITIONED FOR SEGMENT/TEXT POINTER"
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

```
800 PRINT "PRESS  'RETURN' KEY AFTER POSITIONING CURSOR"
810 PRINT "THE TABLE BELOW SHOWS THE LENGTH OF EACH SEGMENT TEXT."
820 PRINT "USE IT AS A GUIDE IN POSITIONING THE CURSOR"
830 IF P9>1 THEN 860
840 PRINT "SEGMENT LABELS WILL BE PRINTED WITHIN THE SEGMENTS"
850 GO TO 870
860 PRINT "SEGMENT LABELS WILL BE PRINTED OUTSIDE THE SEGMENTS"
870 Z=32
880 P2=1
890 FOR I=1 TO N
900     B$=SEG(A$,(I-1)*25+1,25)
910     L=POS(B$,"  ",1)
920     IF L>0 THEN 940
930     L=26
940     PRINT "SEGMENT # ";I;" TEXT LENGTH = ";L
950 NEXT I
960 PRINT "GJJYOU MAY MAKE A HARD COPY OF ABOVE INSTRUCTIONS FOR "
970 PRINT "REFFERENCE. PLEASE NOTE NO FURTHER INSTRUCTIONS WILL"
980 PRINT "APPEAR ON THE SCREEN ONCE THE PLOTTING HAS BEGUNGGG"
990 PRINT "J PRESS `RETURN' WHEN READY TO CONTINUE";
1000 INPUT O$
1010 PRINT "JDO YOU WANT TITLES CENTERED (Y/N) ? ";
1020 INPUT I$
1030 IF I$<>"Y" AND I$<>"N" THEN 1010
1040 Z=32
1050 PAGE @Z:
1060 IF I$="N" THEN 1210
1070 FOR I=1 TO 3
1080     GO TO 1*(I=1)+2*(I=2)+3*(I=3) OF 1090,1110,1130
1090     D$=F$
1100     GO TO 1140
1110     D$=G$
1120     GO TO 1140
1130     D$=H$
1140     X3=36-LEN(D$)/2
1150     FOR I1=1 TO X3
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1160          PRINT @Z:" ";
1170      NEXT I1
1180      PRINT @Z:D$
1190 NEXT I
1200 GO TO 1240
1210 PRINT @Z:F$
1220 PRINT @Z:G$
1230 PRINT @Z:H$
1240 W=1/((W2-E1)/100)
1250 E1=E1/100
1260 WINDOW -W,W,-W,W

1270 REM TELL 4662 TO SCALE CHARACTERS

1280 VIEWPORT 25,125,0,100

1290 REM FIND %

1300 P(1)=0
1310 FOR I=2 TO N+1
1320      P(I)=D(I-1)/S+P(I-1)
1330 NEXT I

1340 REM DRAW REST OF SECTIONS

1350 FOR I=1 TO N
1360      IF N1=99 THEN 1380
1370      IF N1<>I THEN 1430
1380      W3=360*((P(I+1)-P(I))/2+P(I))+180
1390      X=E1*SIN(W3)
1400      Y=E1*COS(W3)

1410      REM SHIFT WINDOW FOR SECTOR BEING SEGREGATED

1420      WINDOW -W+X,W+X,-W+Y,W+Y
1430      MOVE @Z:0,0
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

```
1440      E=360*P(I+1)
1450      X=SIN(E)
1460      Y=COS(E)
1470      DRAW @Z:X,Y
1480      B1=E
1490      FOR J=E TO 360*P(I) STEP -5
1500          DRAW @Z:SIN(J),COS(J)
1510      NEXT J
1520      B2=J
1530      DRAW @Z:SIN(360*P(I)),COS(360*P(I))
1540      DRAW @Z:0,0
1550      X=360*((P(I+1)-P(I))/2+P(I))
1560      Y=X
1570      IF P9=3 THEN 2240

1580      REMARK *   POSITION CURSOR FOR SEGMENT TEXT

1590      B$=SEG(A$,(I-1)*25+1,25)
1600      L=POS(B$,"  ",1)
1610      IF L>0 THEN 1630
1620      L=25
1630      B$=SEG(B$,1,L)
1640      PRINT "GGG"
1650      WINDOW 0,130,0,100
1660      POINTER X1,Y1,Z$
1670      MOVE @Z:X1,Y1
1680      X3=X1
1690      IF S$="%" THEN 1710
1700      B$=B$&S$
1710      IF V$="V" THEN 1790
1720      N$=STR(D(I)/S*100)
1730      L=LEN(N$)
1740      IF L<=5 THEN 1770
1750      N$=SEG(N$,2,5)
1760      GO TO 1820
1770      N$=SEG(N$,2,L-1)
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1780      GO TO 1820
1790      P1=POS(Y$," ",P2)
1800      N$=SEG(Y$,P2,P1-P2)
1810      P2=P1+1
1820      B$=B$&N$
1830      IF S$<>"%" THEN 1850
1840      B$=B$&S$
1850      IF LEN(B$)*2<=130-X1 THEN 2110
1860      P6=1
1870      Q0=0
1880      P7=POS(B$," ",P6)
1890      IF P7>0 THEN 1910
1900      P7=LEN(B$)+1
1910      W$=SEG(B$,P6,P7-P6+1)
1920      IF LEN(W$)*2<=130-X3 THEN 2050
1930      IF X3=X1 THEN 2010
1940      PRINT @Z:"J";
1950      FOR I6=1 TO Q0
1960          PRINT @Z:"H";
1970      NEXT I6
1980      Q0=0
1990      X3=X1
2000      GO TO 1920
2010      FOR I6=1 TO LEN(W$)*2-X1
2020          PRINT @Z:"H";
2030      NEXT I6
2040      INPUT @32,24:X1,Y1
2050      PRINT @Z:W$;
2060      P6=P7+1
2070      X3=X3+LEN(W$)*2
2080      Q0=Q0+LEN(W$)
2090      IF P6=>LEN(B$) THEN 2120
2100      GO TO 1880
2110      PRINT @Z:B$
2120      IF N1<>I THEN 2150
2130      WINDOW -W+E1*SIN(W3),W+E1*SIN(W3),-W+E1*COS(W3),W+E1*COS(W3)
```

TITLE	ABSTRACT NO:
ENHANCED PIE CHART	51/07-9539/0

```
2140      GO TO 2160
2150      WINDOW -W,W,-W,W
2160      IF P9=1 THEN 2240
2170      B3=(B1-B2)/2
2180      X1=SIN(B2+B3)
2190      Y1=COS(B2+B3)
2200      PRINT "GGG"
2210      POINTER X2,Y2,Z$
2220      MOVE @Z:X2,Y2
2230      DRAW @Z:X1,Y1
2240      IF N1=99 THEN 2260
2250      IF N1<>I THEN 2280

2260      REM RESTORE WINDOW

2270      WINDOW -W,W,-W,W
2280 NEXT I
2290 COPY
2300 MOVE @Z:-2,-1.2
2310 RETURN
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
100 REMARK *** PLOT TO PLOTTER "@PIECHR/PLOT4662" 51/07-9539/0

110 PAGE
120 DELETE H1,V1,X7,Y7
130 Z=1
140 DIM H1(10),V1(10),X7(10),Y7(10)
150 FOR I=1 TO 10
160     H1(I)=0
170     V1(I)=0
180     X7(I)=0
190     Y7(I)=0
200 NEXT I
210 VIEWPORT 0,150,0,100
220 WINDOW 0,150,0,100
230 IF U$=" " THEN 250
240 PRINT "YOU HAVE SELECTED ONE-COLOR PLOT"
250 PRINT "JJMOVE PEN TO LOWER LEFT CORNER OF YOUR PLOTTING PAPERGG"
260 PRINT "GG  PRESS `RETURN' KEY WHEN  READY ";
270 INPUT O$
280 INPUT @Z,24:X1,Y1
290 PRINT @Z:"G";
300 PRINT "JMOVE PEN TO UPPER RIGHT CORNER OF YOUR PLOTTING PAPERGG"
310 PRINT "GG  PRESS `RETURN' KEY  WHEN READY ";
320 INPUT O$
330 INPUT @Z,24:X2,Y2
340 PRINT @Z:"G";
350 PRINT "GGGGPLOTTING PEN WILL BE MOVED TO TRACE THE BOUNDARY OF"
360 PRINT "YOUR PLOTTING AREA"
370 PRINT "GGGG   WATCH THE PEN NOW GGGG"
380 FOR J=1 TO 25
390 NEXT J
400 MOVE @Z:X1,Y1
410 PRINT @Z:"G";
420 MOVE @Z:X1,Y2
430 PRINT @Z:"G";
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
 -0 MOVE @Z:X2,Y2
450 PRINT @Z:"G";
460 MOVE @Z:X2,Y1
470 PRINT @Z:"G";
480 MOVE @Z:X1,Y1
490 PRINT "GG DO YOU WISH TO CHANGE PLOTTING AREA (Y/N) :";
500 INPUT I$
510 IF I$<>"Y" AND I$<>"N" THEN 490
520 IF I$="Y" THEN 250
530 X4=X2-X1
540 VIEWPORT X1,X2,Y1,Y2
550 WINDOW 0,X4,0,Y2-Y1
560 IF F$<>" " THEN 600
570 IF G$<>" " THEN 600
580 IF H$<>" " THEN 600
590 GO TO 1120
600 PRINT "JSPECIFY WIDTH AND HEIGHT OF LETTERS FOR THE TITLE"
610 PRINT "ENTER WIDTH (STANDARD IS 1.75) ";
620 INPUT W$
630 IF W$="" THEN 660
640 W1=VAL(W$)
650 IF W1>0 THEN 680
660 W1=1.75
670 PRINT "STANDARD WIDTH WILL BE USED"
680 PRINT "ENTER HEIGHT (STANDARD IS 2.85) ";
690 INPUT W$
700 IF W$="" THEN 730
710 H=VAL(W$)
720 IF H>0 THEN 750
730 H=2.85
740 PRINT "STANDARD HEIGHT WILL BE USED"

750 REMARK CHECK SCALING FACTORS

760 IF LEN(F$)=>LEN(G$) THEN 800
770 IF LEN(H$)=>LEN(G$) THEN 810
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
780 L=LEN(G$)
790 GO TO 840
800 IF LEN(F$)=>LEN(H$) THEN 830
810 L=LEN(H$)
820 GO TO 840
830 L=LEN(F$)
840 X9=(X2-X1)/L
850 H9=X9*2.85/1.75
860 IF X9=>W1 THEN 880
870 W1=X9
880 IF H9=>H THEN 900
890 H=H9
900 PRINT @Z,17:W1,H
910 PRINT "GDO YOU WANT TITLES CENTERED (Y/N) ? ";
920 INPUT I$
930 IF I$<>"Y" AND I$<>"N" THEN 910
940 IF I$="N" THEN 1050
950 X3=(X4-LEN(F$)*W1)/2
960 MOVE @Z:X3,Y2-(H+0.5)
970 PRINT @Z:F$
980 X3=(X4-LEN(G$)*W1)/2
990 MOVE @Z:X3,Y2-(H+0.5)*2
1000 PRINT @Z:G$
1010 X3=(X4-LEN(H$)*W1)/2
1020 MOVE @Z:X3,Y2-(H+0.5)*3
1030 PRINT @Z:H$
1040 GO TO 1110
1050 MOVE @Z:0,Y2-(H+0.5)
1060 PRINT @Z:F$
1070 MOVE @Z:0,Y2-(H+0.5)*2
1080 PRINT @Z:G$
1090 MOVE @Z:0,Y2-(H+0.5)*3
1100 PRINT @Z:H$
1110 PAGE
1120 PRINT "JSEGMENT TEXT AND VALUES MAY BE :"
1130 PRINT " 1  --  PRINTED WITHIN THE SEGMENT"
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1140 PRINT " 2  --  PRINTED OUTSIDE THE SEGMENT WITH A POINTER"
1150 PRINT " 3  --  NOT PRINTED"
1160 PRINT "GSELECT OPTION - 1, 2, OR 3 ? ";
1170 INPUT P$
1180 P9=VAL(P$)
1190 IF P9>0 AND P9<=3 THEN 1220
1200 PRINT "GINVALID RESPONSE --  RE-ENTER ";
1210 GO TO 1170
1220 IF P9=3 THEN 1450
1230 PRINT "JENTER SYMBOL ($,%,...) FOR VALUES: ";
1240 INPUT S$
1250 IF LEN(S$)<=1 THEN 1280
1260 PRINT "ONLY FIRST CHARACTER OF SYMBOL WILL BE PRINTED"
1270 S$=SEG(S$,1,1)
1280 PRINT "JUSE VALUES INPUT (V), OR COMPUTED PERCENTAGE (P)? ";
1290 INPUT V$
1300 IF V$="V" THEN 1430
1310 IF V$="P" THEN 1340
1320 PRINT "GINVALID RESPONSE -- RE-ENTER ";
1330 GO TO 1290
1340 PRINT "GJHOW MANY DECIMAL PLACES FOR THE COMPUTED PERCENTAGE ";
1350 INPUT M$
1360 IF M$<>"" THEN 1390
1370 PRINT "JPERCENTAGE WILL BE PRINTED TO 2 DECIMAL PLACES"
1380 GO TO 1430
1390 M1=VAL(M$)
1400 IF M1<6 THEN 1430
1410 PRINT "GMAXIMUM OF 5 DECIMAL PLACES WILL BE PRINTED"
1420 M1=5
1430 VIEWPORT 0,150,0,100
1440 WINDOW 0,150,0,100
1450 IF P9=3 THEN 1610
1460 PRINT "JSPECIFY WIDTH AND HEIGHT OF LETTERS FOR PIE SEGMENT TEXT "
1470 PRINT "ENTER WIDTH (STANDARD IS 1.75) ";
1480 INPUT W$
1490 IF W$="" THEN 1520
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1500 W1=VAL(W$)
1510 IF W1>0 THEN 1540
1520 W1=1.75
1530 PRINT "GSTANDARD WIDTH WILL BE USED"
1540 PRINT "ENTER HEIGHT (STANDARD IS 2.85) ";
1550 INPUT W$
1560 IF W$="" THEN 1590
1570 H=VAL(W$)
1580 IF H>0 THEN 1610
1590 H=2.85
1600 PRINT "GSTANDARD HEIGHT WILL BE USED"
1610 W2=100
1620 PRINT "JALL SEGMENTS MAY BE SEPARATED OR A SINGLE SEGMENT MAY "
1630 PRINT "SEPARATED FROM THE REST"
1640 PRINT "JSEGMENT SEPARATION OPTIONS ARE :-"
1650 PRINT "  0  --  NO SEPARATION"
1660 PRINT "  n  --  VALID SEGMENT NUMBER"
1670 PRINT "  A  --  ALL SEGEMNT TO BE SEPARATED"
1680 PRINT "JENTER SEGMENT SEPARATION OPTION (0/n/A) ";
1690 INPUT Z$
1700 IF Z$<>"" THEN 1730
1710 PRINT "GINVALID RESPONSE "
1720 GO TO 1680
1730 E1=20
1740 IF Z$<>"A" THEN 1780
1750 PRINT "GA MAXIMUM OF 5% SEGMENT SEPARATION IS RECOMMENDED"
1760 PRINT "FOR SEGMENTS EXPLOSION (OPTION A)"
1770 GO TO 1850
1780 IF ASC(Z$)>47 OR ASC(Z$)<58 THEN 1800
1790 GO TO 1710
1800 N1=VAL(Z$)
1810 IF N1=0 THEN 1940
1820 IF N1<=N THEN 1860
1830 PRINT "GATTEMPTING TO SEPARATE A NON-EXISTENT SEGMENT"
1840 GO TO 1680
1850 N1=99
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
1860 PRINT "JENTER % OF SEPARATION - (RECOMMENDED MAX : 20) ";
1870 INPUT E1
1880 IF E1>0 THEN 1910
1890 PRINT " '0' IS NOT A VALID VALUE -- RE-ENTER ";
1900 GO TO 1870
1910 IF E1<=20 THEN 1940
1920 PRINT "GSEPARATION OF MORE THAN 20% MAY CAUSE LOSS OF "
1930 PRINT "         CORRELATION BETWEEN SEGMENTS"
1940 Z=1
1950 PAGE @Z:
1960 B7=X1
1970 B8=X2
1980 E7=0
1990 E8=0
2000 C$=CHR(95)
2010 PRINT "GJPOSITION PEN AT INTENDED CENTER OF YOUR PIEHHHHHHHHHHHH";
2020 FOR C1=1 TO 30
2030     PRINT "HHHHHHCENTERHHHHHH";C$;C$;C$;C$;C$;C$;"G";
2040 NEXT C1
2050 PRINT
2060 PRINT "   PRESS `RETURN' TO CONTINUE";
2070 INPUT O$
2080 INPUT @Z,24:X0,Y0
2090 PRINT @Z:"G";
2100 PRINT "GJPOSITION PEN AT INTENDED TOP OF YOUR PIEHHHHHHHHHHH";
2110 FOR C1=1 TO 30
2120     PRINT "HHHTOPHHH";C$;C$;C$;"G";
2130 NEXT C1
2140 PRINT
2150 PRINT "     PRESS `RETURN' TO CONTINUE"
2160 INPUT O$
2170 INPUT @Z,24:X6,Y6
2180 Y5=Y6-(X2-X1)*100/150
2190 R=Y6-Y0
2200 IF Y5=>Y1 THEN 2700
2210 IF X2-X1>Y2-Y1 THEN 2230
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
2220 GO TO 2510
2230 Y5=Y6-2*R
2240 IF Y5>Y1 THEN 2260
2250 Y5=Y1

2260 REM SHRINK IN THE X-AXIS

2270 X=(Y6-Y5)*150/100
2280 X1=X0-X/2
2290 X2=X0+X/2
2300 S9=0
2310 F9=12
2320 GO TO 1*(X1<=B7)+2*(X2=>B8) OF 2340,2380
2330 GO TO 2470
2340 S9=S9+1
2350 X1=F9/S9
2360 X2=X1+X
2370 GO TO 2410
2380 S9=S9+1
2390 X2=B8-F9/S9
2400 X1=X2-X
2410 IF P9<>2 THEN 2470
2420 IF S9<4 THEN 2320
2430 PRINT "GGTHERE'S NO ROOM ON THE SIDES OF YOUR PIECHART FOR"
2440 PRINT "SEGMENT LABELS AND VALUES TO BE PRINTEDGGGG"
2450 PRINT "RE-SELECT CENTER AND TOP OF PIE"
2460 GO TO 2010
2470 Y5=Y6-(X2-X1)*100/150
2480 E7=X1
2490 E8=X2
2500 GO TO 2700
2510 PRINT "THE BOTTOM PART OF YOUR PIE WILL BE OFF THE PAGE"
2520 PRINT "IF DRAWN WITH YOUR SPECIFIED CENTER AND RADIUSGG"
2530 PRINT "YOU MAY PROCEED FURTHER WITH ONE OF FOLLOWING OPTIONS"
2540 PRINT "    1. KEEP CENTER AS SPECIFIED & REDUCE RADIUS"
2550 PRINT "    2. MOVE CENTER AND KEEP RADIUS AS SPECIFIED"
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
2560 PRINT "  JSELECT OPTION 1, OR 2 : ";
2570 INPUT C
2580 GO TO C OF 2600,2630
2590 GO TO 2560
2600 Y6=Y6-(Y1-Y5)
2610 Y5=Y1
2620 GO TO 2700
2630 Y6=Y1+2*R
2640 Y5=Y1
2650 MOVE @Z:X6,Y6
2660 PRINT "PEN IS NO NOW AT TOP OF PIE . IS IT OK? (Y/N):";
2670 INPUT R$
2680 GO TO 1*(R$="N")+2*(R$="Y") OF 1950,2626
2690 GO TO 2660

2700 REM *** ADJUST Y-CORDINATE OF LOWER LEFT OF PIE AREA FOR A
2710 REM *** PERFECT CIRCLE

2720 R1=(X2-X1-(Y6-Y5))/2
2730 PRINT "GGGG PLOTTING PEN WILL BE MOVED TO TRACE THE BOUNDARY"
2740 PRINT " OF THE AREA WITHIN WHICH THE PIECHART WILL BE DRAWN"
2750 PRINT "GGGG WATCH THE PLOTTER PEN NOW GGGG"
2760 FOR J=1 TO 25
2770 NEXT J
2780 MOVE @Z:X1+R1,Y5
2790 PRINT @Z:"G";
2800 MOVE @Z:X1+R1,Y6
2810 PRINT @Z:"G";
2820 MOVE @Z:X2-R1,Y6
2830 PRINT @Z:"G";
2840 MOVE @Z:X2-R1,Y5
2850 PRINT @Z:"G";
2860 MOVE @Z:X1+R1,Y5
2870 PRINT "GGGDO YOU WISH TO CHANGE PIECHART AREA? (Y/N) :";
2880 INPUT I$
2890 IF I$<>"Y" AND I$<>"N" THEN 2870
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
2900 IF I$="Y" THEN 2910
2910 R1=(X2-X1-(Y6-Y5))/2
2920 VIEWPORT X1+R1,X2-R1,Y5,Y6
2930 W=1/((W2-E1)/100)
2940 E1=E1/100
2950 WINDOW -W,W,-W,W

2960 REM TELL 4662 TO SCALE CHARACTERS
2970 REM FIND %

2980 P(1)=0
2990 FOR I=2 TO N+1
3000     P(I)=D(I-1)/S+P(I-1)
3010 NEXT I

3020 REM DRAW REST OF SECTIONS

3030 FOR I=1 TO N
3040     IF U$="U" THEN 3070
3050     PRINT "GCOLOR CHANGE FOR NEXT SEGMENT-- PRESS 'RETURN' WHEN READY";
3060     INPUT O$
3070     IF N1=99 THEN 3090
3080     IF N1<>I THEN 3160
3090     W3=360*((P(I+1)-P(I))/2+P(I))+180
3100     X=E1*SIN(W3)
3110     Y=E1*COS(W3)

3120     REM SHIFT WINDOW FOR SECTOR BEING SEGREGATED

3130     WINDOW -W+X,W+X,-W+Y,W+Y
3140     X7(I)=X
3150     Y7(I)=Y
3160     MOVE @Z:0,0
3170     E=360*P(I+1)
3180     X=SIN(E)
3190     Y=COS(E)
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
3200      IF I<>N THEN 3250
3210      IF N1=1 THEN 3250
3220      IF N1=99 OR N1=I THEN 3250
3230      MOVE @Z:A1,A2
3240      GO TO 3260
3250      DRAW @Z:X,Y
3260      B1=E
3270      FOR J=E TO 360*P(I) STEP -2.5
3280          DRAW @Z:SIN(J),COS(J)
3290      NEXT J
3300      B2=J
3310      DRAW @Z:SIN(360*P(I)),COS(360*P(I))
3320      DRAW @Z:0,0
3330      IF I<>1 THEN 3360
3340      A1=SIN(360*P(I))
3350      A2=COS(360*P(I))
3360      X=360*((P(I+1)-P(I))/2+P(I))
3370      Y=X
3380      B3=(B1-B2)/2
3390      H1(I)=SIN(B2+B3)
3400      V1(I)=COS(B2+B3)
3410      IF N1=99 THEN 3430
3420      IF N1<>I THEN 3450

3430      REM RESTORE WINDOW

3440      WINDOW -W,W,-W,W
3450 NEXT I
3460 IF P9=3 THEN 5660

3470 REMARK * ADJUST SCALE FOR SEGMENT TEXT AND VALUE

3480 X9=(X2-X1)/70
3490 IF X9=>W1 THEN 3530
3500 IF H<=2.85 THEN 3520
3510 H=(X2-X1)/70*2.85/1.75
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
3520 W1=X9
3530 PRINT @Z,17:W1,H

3540 REMARK SEGMENT TEXT AND VALUES PRINTING

3550 PRINT @Z,17:W1*W2/100,H*W2/100
3560 PRINT "GLSEGMENT TEXT AND VALUES WILL NOW BE PRINTED"
3570 PRINT "GGGHAVING COME THIS FAR I AM SURE YOU DON'T WANT"
3580 PRINT "TO MESS UP YOUR CHART SO PLEASE FOLLOW INSTRUCTIONS"
3590 PRINT "ON THE SCREEN CLOSELY GGGG"
3600 IF P9>1 THEN 3630
3610 L$="WITHIN"
3620 GO TO 3640
3630 L$="OUTSIDE"
3640 PRINT "SEGMENT LABELS WILL BE PRINTED ";L$;" THE SEGMENTS"
3650 PRINT "JJSEGMENTS ARE NUMBERED FROM 1 TO ";N;" STARTING FROM"
3660 PRINT "THE 12 O'CLOCK POSITION AND COUNTING CLOCKWISE"
3670 Z2=1
3680 FOR I=1 TO N
3690     B$=SEG(A$,(I-1)*25+1,25)
3700     L=POS(B$,"  ",1)
3710     IF L>0 THEN 3740
3720     B$=B$&"  "
3730     L=26
3740     B$=SEG(B$,1,L)
3750     IF S$="%" THEN 3770
3760     B$=B$&S$
3770     IF V$="P" THEN 3820
3780     Z1=POS(Y$," ",Z2)
3790     N$=SEG(Y$,Z2,Z1-Z2)
3800     Z2=Z1+1
3810     GO TO 4030
3820     N9=D(I)/S*100
3830     GO TO M1+1 OF 3840,3860,3880,3900,3920,3940
3840     N9=N9+0.5
3850     GO TO 3950
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51/07-9539/0

```
3860     N9=N9+0.05
3870     GO TO 3950
3880     N9=N9+0.005
3890     GO TO 3950
3900     N9=N9+5.0E-4
3910     GO TO 3950
3920     N9=N9+5.0E-5
3930     GO TO 3950
3940     N9=N9+5.0E-6
3950     N$=STR(N9)
3960     L=POS(N$,".",1)
3970     IF M1<>0 THEN 4000
3980     N$=SEG(N$,1,L-1)
3990     GO TO 4020
4000     IF LEN(N$)-L<=M1 THEN 4020
4010     N$=SEG(N$,1,L+M1)
4020     N$=SEG(N$,2,LEN(N$)-1)
4030     B$=B$&N$
4040     IF S$<>"%" THEN 4060
4050     B$=B$&S$
4060     L=LEN(B$)
4070     MOVE @Z:H1(I),V1(I)
4080     PRINT "JGPEN POSITION INDICATES MID-POINT OF ARC FOR"
4090     PRINT "SEGMENT ";I;". USE IT AS A GUIDE TO POSITION"
4100     PRINT "PEN FOR SEGMENT LABEL ";L;
4110     PRINT " CHARACTERS LONG"
4120     PRINT "G              PRESS `RETURN' WHEN READY";
4130     INPUT O$
4140     VIEWPORT X1,X2,Y1,Y2
4150     WINDOW 0,X2-X1,0,Y2-Y1
4160     GIN @Z:X9,Y9
4170     X8=X2-X9-X1
4180     IF X8>(X2-X1)/2 THEN 4820
4190     IF X8=>L*W1 THEN 4990
4200     FOR I1=1 TO LEN(B$)-INT(X8/W1)
4210         PRINT @Z:"H";
```

TITLE	ABSTRACT NO.
ENHANCED PIE CHART	51'07-9539/0

```
4220     NEXT I1
4230     PRINT "GPEN'S BEEN POSITIONED TO FIT LABEL ON 1 LINEG"
4240     PRINT "OK TO PRINT? (Y?N) ";
4250     INPUT I$
4260     IF I$<>"Y" AND I$<>"N" THEN 4240
4270     IF I$="Y" THEN 4990
4280     I3=0
4290     X$=""
4300     P2=1
4310     P1=POS(B$," ",P2)
4320     IF P1>0 THEN 4370
4330     P1=LEN(B$)
4340     IF P1-P2+1<LEN(X$) THEN 4430
4350     W$=SEG(B$,P2,P1-P2+1)
4360     GO TO 4430
4370     T$=SEG(B$,P2,P1-P2+1)
4380     P2=P1+1
4390     I3=I3+1
4400     IF LEN(X$)>LEN(T$) THEN 4310
4410     X$=T$
4420     GO TO 4310
4430     IF X$=B$ THEN 5370
4440     FOR I1=1 TO LEN(B$)-LEN(X$)
4450         PRINT @Z:" ";
4460     NEXT I1
4470     PRINT "GPEN'S BEEN POSITIONED TO PRINT SEGMENT LABEL ON ";
4480     PRINT "MULTIPLE LINES."
4490     PRINT "GIF OK TO PRINT PRESS `RETURN' "
4500     PRINT "GIF NOT, REPOSITION PEN AND PRESS `RETURN' ";
4510     INPUT O$
4520     GIN @Z:X9,Y9
4530     IF LEN(X$)*W1<X2-X9-X1 THEN 4630
4540     PRINT "GGLONGEST `WORD' IN LABEL WILL NOT FIT INTO AVAILABLE";
4550     PRINT " SPACE"
4560     PRINT "GMOVE PEN BACK AND ENTER `Y'"
4570     PRINT "GGG        OR"
```

TITLE	ABSTRACT NO:
ENHANCED PIE CHART	51/07-9539/0

```
4580     PRINT "GENTER `N' ";
4590     INPUT I$
4600     IF I$<>"Y" AND I$<>"N" THEN 4560
4610     IF I$="Y" THEN 4520
4620     GO TO 5370
4630     L1=0
4640     P2=1
4650     FOR I1=1 TO I3+1
4660         P1=POS(B$," ",P2)
4670         IF P1>0 THEN 4690
4680         P1=LEN(B$)
4690         T$=SEG(B$,P2,P1-P2+1)
4700         P2=P1+1
4710         GIN @Z:X9,Y9
4720         IF X2-X9-X1=>LEN(T$)*W1 THEN 4780
4730         PRINT @Z:"J";
4740         FOR I2=1 TO L1
4750             PRINT @Z:"H";
4760         NEXT I2
4770         L1=0
4780         PRINT @Z:T$;
4790         L1=L1+LEN(T$)
4800     NEXT I1
4810     GO TO 5510
4820     FOR I2=1 TO LEN(B$)
4830         PRINT @Z:" ";
4840     NEXT I2
4850     PRINT "GPEN POSITION INDICATES WHERE SEGMENT LABEL WILL END"
4860     PRINT "WHEN PRINTED ON 1 LINE."
4870     PRINT "     ENTER :I`P' TO PRINTIIII`M' TO PRINT ON MULTIPLE LINE";
4880     PRINT "SIII`R' TO REPOSITION PEN MANUALLY ";
4890     INPUT I$
4900     GO TO 1*(I$="P")+2*(I$="M")+3*(I$="R") OF 4960,5010,4930
4910     PRINT "GINVALID RESPONSE -- RE-ENTER ";
4920     GO TO 4890
4930     PRINT "GREPOSITION PEN AND PRESS `RETURN' ";
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
4940     INPUT O$
4950     GO TO 4160
4960     FOR I2=1 TO LEN(B$)
4970         PRINT @Z:"H";
4980     NEXT I2
4990     PRINT @Z:B$;
5000     GO TO 5510
5010     PRINT "GBEFORE EACH `WORD' IN SEGMENT LABEL IS PRINTED, THE"
5020     PRINT "PEN WILL BE POSITIONED TO INDICATE WHERE THE `WORD' "
5030     PRINT "WILL END WHEN PRINTED. YOU WILL BE REQUIRED TO INDICATE"
5040     PRINT "IF `WORD' SHOULD BE PRINTED ON THE CURRENT LINE OR PEN"
5050     PRINT "SHOULD SKIP DOWN TO THE BEGINNING OF NEXT LINE"
5060     FOR I2=1 TO LEN(B$)
5070         PRINT @Z:"H";
5080     NEXT I2
5090     P2=1
5100     L1=0
5110     P1=POS(B$," ",P2)
5120     IF P1>0 THEN 5140
5130     P1=LEN(B$)
5140     T$=SEG(B$,P2,P1-P2+1)
5150     FOR I2=1 TO LEN(T$)
5160         PRINT @Z:" ";
5170     NEXT I2
5180     PRINT "GOK TO PRINT (Y/N)? ";
5190     INPUT I$
5200     IF I$<>"Y" AND I$<>"N" THEN 5180
5210     IF I$="N" THEN 5260
5220     FOR I2=1 TO LEN(T$)
5230         PRINT @Z:"H";
5240     NEXT I2
5250     GO TO 5320
5260     PRINT @Z:"J";
5270     FOR I2=1 TO L1+LEN(T$)
5280         PRINT @Z:"H";
5290     NEXT I2
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
 300     L1=0
5310     GO TO 5150
5320     PRINT @Z:T$;
5330     L1=L1+LEN(T$)
5340     IF P1=LEN(B$) THEN 5510
5350     P2=P1+1
5360     GO TO 5110
5370     PRINT "GONLY ";INT((X2-X9-X1)/W1);" CHARACTERS WILL FIT"
5380     PRINT "INTO AVAILABLE SPACE"
5390     PRINT "RE-ENTER SEGMENT LABEL, INSERTING A DASH `-' AND ONE "
5400     PRINT "BLANK WHEREVER YOU WISH TO BREAK TO A NEW LINEGG"
5410     PRINT "JLABEL TO BE PRINTED IS: ";B$
5420     DELETE B$
5430     INPUT B$
5440     I3=0
5450     P2=1
5460     P1=POS(B$," ",P2)
5470     IF P1=0 THEN 4630
5480     P2=P1+1
5490     I3=I3+1
5500     GO TO 5460
5510     IF P9=1 THEN 5550
5520     PRINT "GPOSITION PEN FOR LABEL===>SEGMENT POINTER "
5530     PRINT "      PRESS `RETURN' WHEN READYG";
5540     INPUT O$
5550     IF E8=0 THEN 5580
5560     X1=E7
5570     X2=E8
5580     VIEWPORT X1+R1,X2-R1,Y5,Y6
5590     IF N1=99 OR N1=I THEN 5620
5600     WINDOW -W,W,-W,W
5610     GO TO 5630
5620     WINDOW -W+X7(I),W+X7(I),-W+Y7(I),W+Y7(I)
5630     IF P9=1 THEN 5650
5640     DRAW @Z:H1(I),V1(I)
5650 NEXT I
```

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

5660 RETURN

TITLE	ABSTRACT NO
ENHANCED PIE CHART	51/07-9539/0

```
100 REMARK *** SAVE PLOT DATA "@PIECHR/SAVEPLOT" 51/07-9539/0

110 PRINT "LDO YOU WANT CHART SAVED ON DISK OR TAPE? (D/T) :";
120 INPUT M$
130 M$=SEG(M$,1,1)
140 GO TO 1*(M$="T")+2*(M$="D")+3*(M$="") OF 190,260,170
150 PRINT "GINVALID RESPONSE -- RE-ENTER ";
160 GO TO 120
170 PRINT "GJYOUR CHART WILL NOT BE SAVED"
180 RETURN
190 IF Z8=0 THEN 210
200 PRINT "JJTHIS DATA WAS READ FROM FILE ";Z8
210 PRINT "JJINSERT TAPE AND ENTER FILE NUMBER TO SAVE CHART ";
220 INPUT Z8
230 FIND Z8
240 WRITE @33:A$,F$,G$,H$,D,N,S,Y$
250 RETURN
260 IF J$="" THEN 290
270 J$=SEG(J$,9,LEN(J$)-8)
280 PRINT "JTHIS DATA WAS READ FROM FILE - ";J$
290 PRINT "JUSER NAME AND FILE ARE LIMITED TO 10 CHARACTERS"
300 PRINT "ENTER (User name/File name) ? ";
310 INPUT J$
320 J$="@PIECHR/"&J$
330 CALL "FILE",0,J$,Z$
340 IF Z$<>"" THEN 360
350 CREATE J$;1000,0
360 OPEN J$;1,"F",Z$
370 WRITE #1:A$,F$,G$,H$,D,N,S,Y$
380 CLOSE 1
390 DELETE J$
400 PRINT "JJJ   PLOT DATA SAVED . . ."
410 RETURN
```

ENHANCED PIE CHART

ABSTRACT NO

51/07-9539/0

```
100 REMARK *** DRAW BOX AROUND PIECHART "@PIECHR/BORDER" 51/07-9539/0

110 PAGE
120 PRINT "GMOVE PEN TO LOWER LEFT CORNER OF PAGE BORDER"
130 PRINT "       PRESS `RETURN' KEY WHEN READYGG";
140 INPUT O$
150 Z=1
160 INPUT @Z,24:X1,Y1
170 PRINT @Z:"G";
180 PRINT "JGMOVE PEN TO UPPER RIGHT CORNER OF PAGE BORDER"
190 PRINT "        PRESS `RETURN' KEY WHEN READYGG"
200 INPUT O$
210 INPUT @Z,24:X2,Y2
220 PRINT @Z:"G";
230 X3=X2-X1
240 Y3=Y2-Y1
250 VIEWPORT X1,X2,Y1,Y2
260 WINDOW 0,X3,0,Y3
270 MOVE @Z:0,0
280 RDRAW @Z:0,Y3
290 RDRAW @Z:X3,0
300 RDRAW @Z:0,-Y3
310 RDRAW @Z:-X3,0
320 PRINT "GJDO YOU WISH TO REPEAT PAGE BORDER (Y/N) ? ";
330 INPUT I$
340 IF I$<>"Y" AND I$<>"N" THEN 320
350 IF I$="Y" THEN 280
360 VIEWPORT 0,150,0,100
370 WINDOW 0,150,0,100
380 RETURN
```